Bagging deep convolutional autoencoders trained with a mixture of real data and GAN-generated data

Cong Hu[1], Xiao-Jun Wu[*1], Zhen-Qiu Shu[1, 2]

[1]School of Internet of Things Engineering, Jiangnan University,
Wuxi 214122, Jiangsu Province, China

[2]School of Computer Engineering, Jiangsu University of Technology,
Changzhou 231001, Jiangsu Province, China

[e-mail: wxhucong@163.com, jnu_xiaojun_wu@163.com, shuzhenqiu@163.com]

*Corresponding author: Xiao-Jun Wu

Received March 28, 2018; revised October 29, 2018; revised January 22, 2019; accepted June 18, 2019; published November 30, 2019

Abstract

While deep neural networks have achieved remarkable performance in representation learning, a huge amount of labeled training data are usually required by supervised deep models such as convolutional neural networks. In this paper, we propose a new representation learning method, namely **g**enerative **a**dversarial **n**etworks (GAN) based **b**agging **d**eep **c**onvolutional **a**uto**e**ncoders (GAN-BDCAE), which can map data to diverse hierarchical representations in an unsupervised fashion. To boost the size of training data, to train deep model and to aggregate diverse learning machines are the three principal avenues towards increasing the capabilities of representation learning of neural networks. We focus on combining those three techniques. To this aim, we adopt GAN for realistic unlabeled sample generation and bagging deep convolutional autoencoders (BDCAE) for robust feature learning. The proposed method improves the discriminative ability of learned feature embedding for solving subsequent pattern recognition problems. We evaluate our approach on three standard benchmarks and demonstrate the superiority of the proposed method compared to traditional unsupervised learning methods.

This research was supported by the 111 Project of Chinese Ministry of Education under Grant B12018, the National Nature Science Foundation of China [Grant 61373055, 61672265 and 61603159] and the Natural Science Foundation of Jiangsu Province of China under Grant BK20160293.

http://doi.org/10.3837/tiis.2019.11.009 **ISSN : 1976-7277**

Keywords: representation learning; unsupervised learning; generative adversarial networks; deep convolutional autoencoders; bagging

1. Introduction

Deep neural networks (DNNs) have progressed rapidly in recent years and have been applied successfully to many computational tasks, including speech recognition, natural language processing (NLP), information retrieval, computer vision, and image analysis [1-7]. In the fields of computer vision, the most relevant procedures of the winners follow three main avenues: extending the training data, building ensembles of learning machines, and constructing DNNs.

Supervised learning methods such as convolutional neural networks (CNN), however, require large quantities of labeled data that can be difficult to obtain. Thus, building a robust system that uses unsupervised learning is valuable. Unsupervised learning methods determine intrinsic representations that preserve the essential aspects of unlabeled data. Such methods are typically used to extract useful information and remove redundancies in the raw data. Some meaningful features like image edges and color can be extracted from raw images with the unsupervised learning methods such as spatial pyramid matching [8] and bag of visual words [9]. These shallow models can be stacked to form a deep model that is capable of extracting abstract features like contours [10, 11].

In recent years, CNN [12, 13] has become one of the most successful deep models for automatically extracting the hierarchical and discriminative features in an end-to-end training manner. CNNs show superior performance on visual recognition problems in particular. The success of CNN comes from its deep structure and the use of massive training data, which allow CNN to learn meaningful hierarchical representations and improve the performance of subsequent recognition tasks [14, 15]. However, a large number of labeled data are usually expensive to obtain, and the expense limits wider use of this supervised deep model.

As a result, explorations of unsupervised methods to learn hierarchical and intrinsic representations from the unlabeled data is valuable. Stacked autoencoders (SAEs) [16, 17] is one of the typical deep models used for unsupervised feature learning. It stacks shallow autoencoders to form a deep model that learns the latent representations in the input by

reconstructing itself layer by layer. Shin et al. [18] applied stacked sparse autoencoders (SSAEs) to recognize medical images with outstanding improvements in recognition performance. Vincent et al. [19, 20] introduced denoising autoencoders (DAE) that learns the features in noisy data by adding random noise to the input layers and reconstructing the clean ones at the training stage. DAE can also be stacked to form a deep unsupervised network for learning deep representations. Masci et al. [21] presented stacked convolutional autoencoders (SCAE) to initialize the weights of the deep models and achieved excellent performance. In view of the success of these deep models and convolutional architectures, we propose deep convolutional autoencoders (DCAE) to quickly predict latent feature maps.

The key to the success of deep feature learning is big data. Many classifiers use techniques to increase the number of training examples. Date augmentation often serves as a regularizer to prevent overfitting in deep learning. An automatic image generator can be constructed from generative adversarial networks (GANs) [22]. GANs learn the manifold structure of the data and is able to generate realistic data that has the same distribution as the real one in an unsupervised manner. GAN-generated data can augment the training dataset and regularize the discriminative feature learning. When the deep features are learned from the boosted training dataset and the deep convolutional autoencoders, the whole model shows improved performance for subsequent recognition tasks.

Almost all of the winning solutions in the ImageNet Large Scale Visual Recognition Challenge [23] from 2012 to 2017 are ensembles of CNNs. Bootstrap Aggregation (Bagging) [24-26] is the best-known method in the independent ensemble framework. Bagging reduces variance and model variability over different data sets from a given distribution, which reduces the overall generalization error and improves stability as demonstrated in numerous published studies. Bagging creates several bootstrapping subsets of the training dataset and then employs these subsets to train separate models. Some of the original samples may not appear at all and some others may appear more than once since the data are sampled with replacement. After each individual prediction is obtained, the bagging method combines them using a voting scheme to make the final recognition. As these training subsets are slightly different from each other, different focus and parameters are trained on different subsets and thereby receive different prediction errors. Alternatively, each learning machine is built independently. By combining these individuals together, we expect an improved performance of the learning machine and a decrease in the total prediction error. In the real world, a human being usually considers multiple possibilities before making a final decision. We weigh these individual possibilities and combine them to make the final decision.

In previous work, the bagging method performed well for unstable predictors. Autoencoder-based prediction models are also unstable predictors, it is intuitive to assume that applying the bagging method to autoencoder-based models could improve classification performance. Inspired by this assumption, we propose a bagging deep convolutional autoencoder-based (BDCAE) prediction architecture for robust feature learning. The

experimental results on commonly used datasets also show its promising potential. We adopt a bagging strategy to fuse multiple deep convolutional autoencoders to reduce the generalization error, thus improving the classification accuracy further. We summarize our contributions as follows:

(1) We propose a novel framework, GAN-BDCAE, for robust and discriminative feature learning in an unsupervised fashion.

(2) The integration of additionally generated data regularizes the process of discriminative feature learning. Experiments on image classification tasks validate the effectiveness of the GAN-generated data for improving the generalization capability of the learned features.

(3) A demonstration that the learned feature of the proposed method GAN-BDCAE has a consistent improvement over three image benchmark datasets when compared with other traditional feature learning methods. It is also a flexible and effective framework for semi-supervised learning when there are at least a few labeled data items.

The rest of this paper is organized as follows: in Section 2, we present the proposed GAN-BDCAE framework. We show our experimental results in Section 3 and draw a conclusion in Section 4.

2. The Proposed Method

2.1 Architecture overview

Image representation plays a key role in image recognition tasks [27]. To improve the ability of image representation, we propose a new method that uses three techniques. The first technique improves the diversity of the training data with a GAN. The second technique learns the hierarchical representation of the augmented training dataset with deep convolutional autoencoders (DCAE). The third technique boosts the robustness of predictors by bagging. The architecture of our proposed method GAN-BDCAE is shown in **Fig. 1**. During the unsupervised feature learning step, we train our model to obtain the data structure and improve the generalizability of the method. We then fine-tune the whole model with N labeled real data and assemble them with bagging to achieve a robust recognition system.

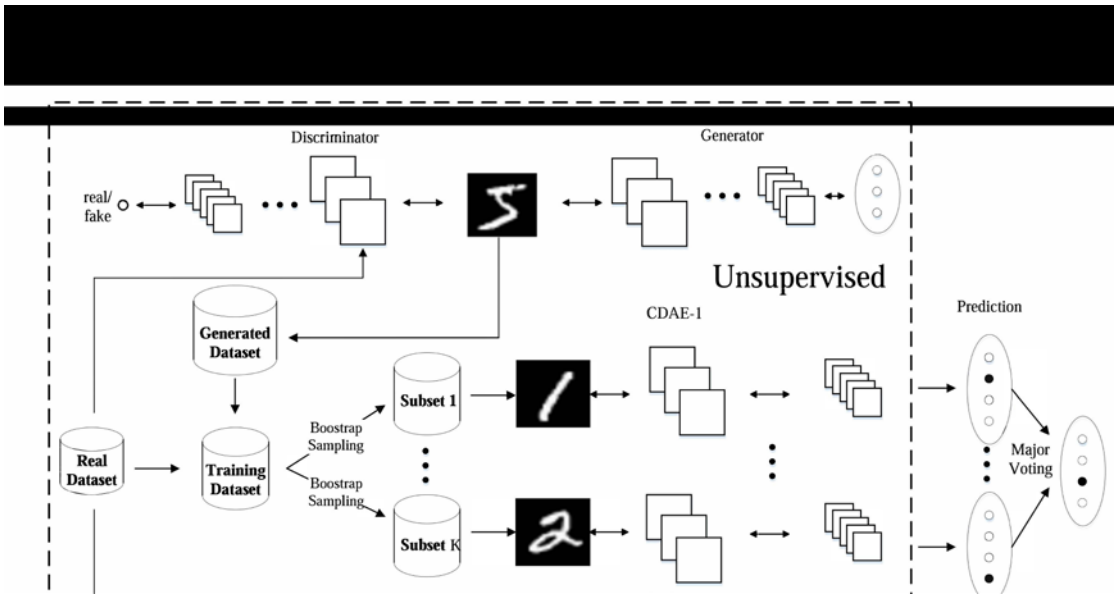


Fig. 1. Architecture of our proposed method GAN-BDCAE

2.2 Data augmentation using GAN

The GAN consists of two sub-networks: a generator and a discriminator. The discriminator determines whether a sample is generated or real while the generator produces samples to deceive the discriminator. Goodfellow et al. [28] first proposed GANs to generate images and gain insight into neural networks. The deep convolutional GAN adds refinements to improve training stability. The discriminator of Deep Convolutional GAN can serve as a robust feature extractor. Salimans et al. [29] achieved state-of-the-art results for semi-supervised classification and improved the visual quality of GANs. InfoGAN [30] learns interpretable representations by introducing latent codes. GANs also demonstrate potential in generating images for specific applications. Pathak et al. [31] proposed an encoder-decoder method for image inpainting (reconstructing missing or deteriorated parts of an image) using GANs to generate the images. Similarly, Yeh et al. [32] improved inpainting performance by introducing two loss types. Our own aim in this work is to use a deep convolutional GAN model to generate realistic unlabeled samples and to show that these samples improve discriminative feature learning.

2.3 Deep convolutional autoencoders

One of our aims is to use generated data to improve discrimination when learning features. Using deep convolutional autoencoders as a feature learning machine, we design the deep convolutional autoencoders as convolutional stacked of deep encoder and decoder layers.

The basic autoencoder component, with its encoder-decoder structure, has been widely used as an unsupervised feature learning tool in other research. In the encoding phase, the

machine transforms the input space into an application-specific latent space. The decoding phase uses this latent representation to reproduce the original data [33]. To encode data as robust and discriminative representations, we train the autoencoder to extract generally useful features, to remove redundancies in the inputs, and to preserve essential properties of the input data. The mathematical representation of the autoencoders is given by

$$\hat{x}=f_{w,b}(x)\approx x \tag{1}$$

where $x\in[0,1]^n$ represents the input data and $W=\{W_1,W_2\}$ and $b=\{b_1,b_2\}$ denote the connecting weights and the layer biases respectively. First, the autoencoder maps x to the hidden representation h through an encoder mapping parameterized by $\theta_1=\{W_1,b_1\}$ and defined as

$$h=g_{\theta_1}(x)=\sigma(W_1x+b_1) \tag{2}$$

where $h\in[0,1]^{n'}$, $W_1\in R^{n'\times n}$, $b_1\in R^{n'\times 1}$ and $\sigma(x)$ is the logistic sigmoid function, $\sigma(x)=\frac{1}{1+e^{-x}}$. A similar decoder mapping function parameterized by $\theta_2=\{W_2,b_2\}$ maps the hidden representation h back to a reconstructed vector $\hat{x}\in[0,1]^n$:

$$\hat{x}=g_{\theta_2}(x)=\sigma(W_2h+b_2) \tag{3}$$

where $W_2\in R^{n\times n'}$ and $b_2\in R^{n\times 1}$. Basic autoencoders training consists of finding parameters of the model in Eq. (1), that is, $\theta=\{\theta_1,\theta_2\}$. In order to optimize these parameters, the objective function of the autoencoders is to minimize the average reconstruction error.

However, the fully connected AEs ignore the 2-D image structure. Since the inputs are images, it makes sense to use convolutional networks for encoding and decoding. In practical settings, autoencoders applied to images are almost always convolutional autoencoders (CAEs) for the sake of performance. CAEs are quite similar to conventional AEs but differ by having their weights shared among all locations in the input to preserve spatial locality. For a mono-channel input x, the latent representation of the j-th feature map is given by

$$h^j=\sigma(x*W_1^j+b_1^j) \tag{4}$$

where the bias b_1 is broadcast to the whole map, σ is an activation function (rectified linear units in our experiments), and * denotes the 2D convolution. We use a single bias per latent map, as we want each filter to specialize on features of the whole input (One bias per pixel would introduce too many degrees of freedom for a fully connected AE). The reconstruction is obtained using

$$\hat{x} = \sigma(\sum_{j \in H} h^j * W_2^j + b_2) \tag{5}$$

where again there is one bias per input channel. *H* denotes the group latent feature maps. The cost function to be minimized is the mean squared error (MSE):

$$J(W,b) = \frac{1}{2n}\sum_{i=1}^{n} \| \hat{x}^{(i)} - x^{(i)} \|^2 \tag{6}$$

where *i* denotes the *i*-th sample and *n* is the total number of the training data. The weights and bias can be updated using the stochastic gradient descent method.

2.4 Bagging the DCAE-based classifiers

First, the bagging step processes the training dataset (consisting of real data and realistic GAN-generated data) by bootstrapping it. Given a dataset, the system builds bootstrap subsets by randomly sampling the new training dataset with replacement. Because of the replacement sampling strategy, some data may not be picked at all and others may be picked more than once. Each bootstrap subset will contain about 67% of the total training data. Next, each bootstrap subset is used to train a deep convolutional autoencoder-based classifier. At the conclusion, there are *k* DCAE-based classification models with different initial training weights used to initialize the individual neural networks. The outputs of the *k* models are all potentially different. The final model output is computed by a majority vote within the recognition task.

Once the *k* DCAEs are trained on the *k* different subsets, all information will be stored in the parameters of each DCAE. We train a softmax regression model based on top of the coder of each DCAE for the multi-class prediction. The class label *y* takes more than two values, where $y \in 1,2,...,c$ and *c* is the number of the class labels. For an example *x*, we estimate the probabilities of each class that *x* belongs to as follows:

$$h_\theta(x) = \begin{bmatrix} p(y_i = 1 \mid x;\theta) \\ p(y_i = 2 \mid x;\theta) \\ \vdots \\ p(y_i = c \mid x;\theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{c} e^{\theta_j^{\mathrm{T}} x}} \begin{bmatrix} e^{\theta_1^{\mathrm{T}} x} \\ e^{\theta_2^{\mathrm{T}} x} \\ \vdots \\ e^{\theta_c^{\mathrm{T}} x} \end{bmatrix} \tag{7}$$

where $\sum_{j=1}^{c} e^{\theta_j^{\mathrm{T}} x}$ is a normalized term, and $\theta_1, \theta_2, ..., \theta_c$ are the model parameters.

Given the labeled training set $\{x_i, y_i\}_{i=1}^{N}$, $y_i \in 1, 2, ..., c$, the solution of the softmax regression is obtained by minimizing the following optimization problem:

$$\min_{\theta} \left(-\frac{1}{N} \sum_{i=1}^{n} \sum_{j=1}^{c} I\{y_i = j\} \log \frac{e^{\theta_j^{\mathrm{T}} x_i}}{\sum_{l=1}^{c} e^{\theta_l^{\mathrm{T}} x_i}} \right) \tag{8}$$

where $I\{\bullet\}$ is an indicator function with a value of 1 if the expression is true, and 0 otherwise. Once the model is trained, we compute the probability of sample *x* belonging to a label *j* using Eq. (7) and assign its class label via

$$y = \max_{\theta} \frac{e^{\theta_j^{\mathrm{T}} x_i}}{\sum_{l=1}^{c} e^{\theta_l^{\mathrm{T}} x_i}} \tag{9}$$

We obtain the *k* hypothesis function $p_1, p_2, ..., p_k$ to represent the *k* predictors and give a prediction. Each trained model has identified different features of interest. So the predicted labels $p_1(x), p_2(x), ..., p_k(x)$ will not always be the same. In this paper, we set *k* to 3. For a possible label *y* within the set *Y* of all possible labels, a hypothesis function p_i for prediction model *i*, and combined prediction function $p*$ we combine the prediction in the major voting process according to

$$p*(x) = \arg\max_{y \in Y} \sum_{i: p_i(x) = y} 1 \tag{10}$$

2.5 Implementation Details

We have implemented the proposed framework GAN-BDCAE using the Tensorflow and Keras libraries. We used Tensorflow for generating the GAN-based data and Keras for feature learning, subsequent prediction tasks, and bagging. The framework includes six steps as follows:

(1) Generate realistic dataset B^G with deep convolutional GAN.

$B^G = \{x_j^\dagger\}$, $j = 1, 2, ..., m$: A set of generated realistic data.

(2) Form a new training dataset *B* combining the generated data B^G and real data B^R.

$B = \{B^G, B^R\} = \{x_i, x_j^\dagger\}$, $i = 1, 2, ..., n$: A new training dataset which consists of real data x_i and generated data $x_j^\dagger$.

(3) Generate *k* bootstrap subsets with a mixture of real data and GAN-generated data.

B_i, $i = 1, 2, ..., k$: A set of bootstrapping subsets from the augmented new training data set *B*.

(4) Train *k* different deep convolutional autoencoders with Eq. (6) for feature learning on the *k* bootstrap subsets.

$g_i(x)$, $i = 1, 2, ..., k$, $x \in B_i$: Train DCAE-1, DCAE-2, …, DCAE-*k* based on the *k* different bootstrap subset B_i.

(5) Using the features learned with DCAE, train the *k* different classifiers with *N* real labeled samples from the real dataset. Then fine-tune the *k* different DCAE-based classification model.

$f_i(x, y)$, $i = 1, 2, ..., k$, $(x, y) \in B^R$: Train the *k* classifiers and fine-tune each DCAE-based classifier with *N* labeled samples.

(6) Aggregate *k* outputs for the final prediction.
Aggregate the outputs of learners based on majority voting by using Eq. (10).

3. Experiments

3.1 Data augmentation by GAN

In this subsection, we present our experiment using a GAN-based generator to synthesize realistic data for comparison with real data from three benchmark datasets: MNIST [34], SVHN [35] and CIFAR-10 [36]. We synthesized new images by inputting 100-dimensional random vectors in which each entry falls within [-1, 1].

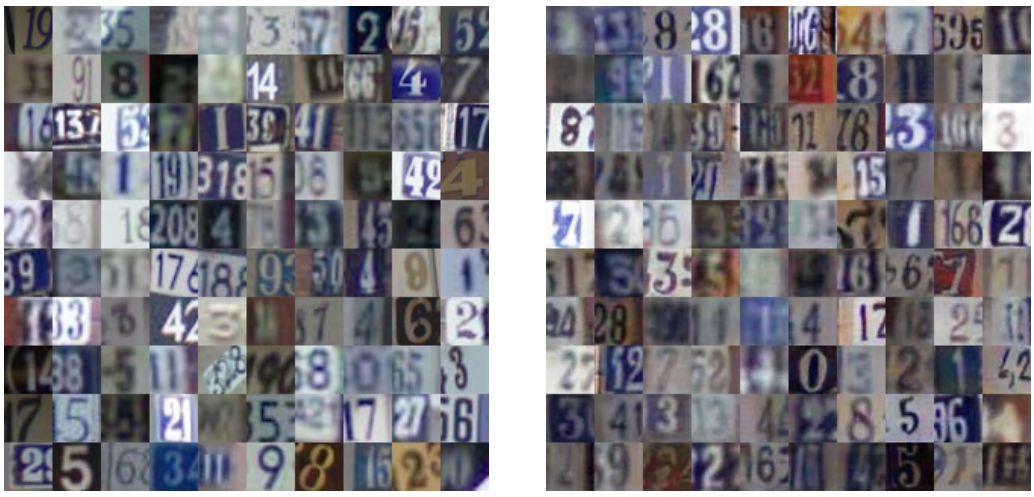
MNIST is a well-known handwritten digit dataset. In the first experiment, we trained the proposed method on the standard benchmark dataset MNIST. This dataset contains digits 0 to 9(10-classes), consisting of 28*28 pixel black and white images. There are 60,000 training images and 10,000 test images. Before we input these images to our model, we

scaled the pixel values into the range [0, 1]. **Fig. 2** provides samples of both the original MNIST and GAN-generated data.

(a) Original images (b) Generated images

Fig. 2. Samples of (a) original images and (b) GAN-generated images from the MNIST dataset

SVHN is a real-world dataset for evaluating image recognition performance, with 73,257 training points and 26,032 test points. **Fig. 3** show samples of actual SVHN and GAN-generated data.



(a) Original images (b) Generated images

Fig. 3. Samples of (a) some original images and (b) GAN-generated images from the SVHN dataset

CIFAR-10 is an established computer vision dataset used for testing object recognition. It consists of 60,000 32*32 color images containing 1-10 object classes, with 6,000 images per class. The CIFAR-10 dataset is split into six batches, each with 10,000 images containing about 1,000 randomly-selected images from each image class. We separated the CIFAR-10 dataset into five batches for training and one batch for testing. **Fig. 4** compares sample CIFAR-10 and GAN-generated images.

(a) Original images (b) Generated images

Fig. 4. Samples of (a) original images and (b) GAN-generated images from the CIFAR-10 dataset

3.2 Deep feature learning

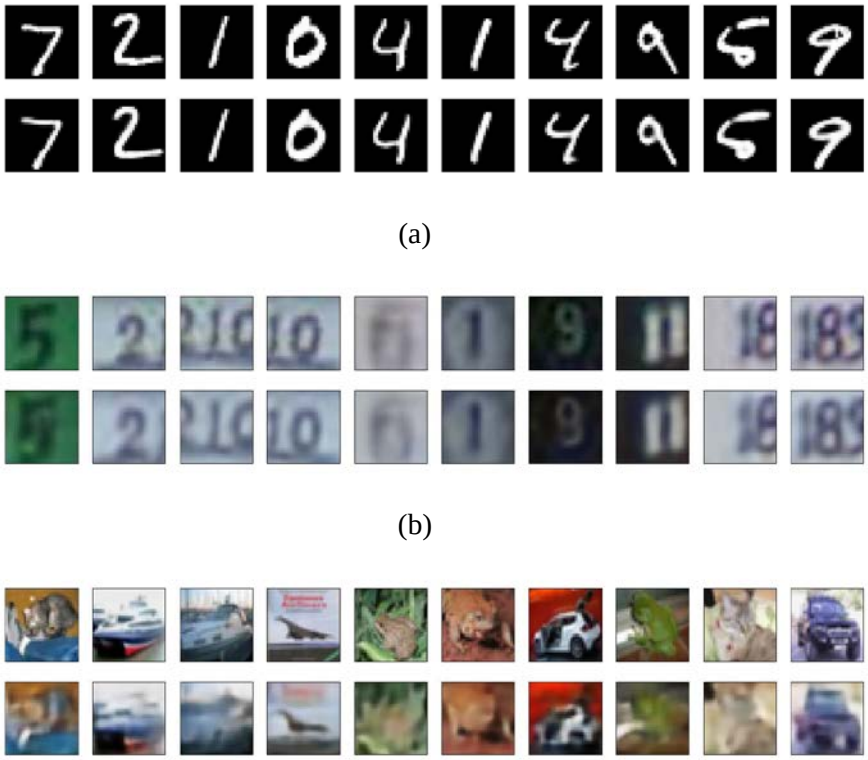
To test deep feature learning, we fed generated data into the GAN-BDCAE model and used the new training dataset to learn deep features. We configured the GAN-BDCAE with k individuals. The k DCAEs had the same architecture but different initial parameters and training subsets. The feature learning machine used deep convolutional autoencoders. The encoder consisted of a stack of convolutional and max pooling layers (for spatial downsampling), while the decoder included a stack of convolutional and upsampling layers. Table 1 and 2 show details of our DCAE models. The DCAE in **Table 1** was used for deep feature learning with the MNIST dataset, while the DCAE in **Table 2** was used with the SVHN and CIFAR-10 dataset. We then trained these DCAEs for latent representations and reproduced the input data. **Fig. 5** compares some original images and their corresponding reconstructions from all 3 datasets.

Table 1. Architecture of DCAE on the MNIST dataset

Stage	Layer	Layer Type	Size	Output Shape
	0	Input		(1,28,28)
	1	Convolution+ReLU	16,3×3filters	(16,28,28)
	1	Max Pooling	2×2,stride2	(16,14,14)
Encoder	2	Convolution+ReLU	32,3×3filters	(32,14,14)
	2	Max Pooling	2×2,stride2	(32,7,7)
	3	Convolution+ReLU	64,3×3filters	(64,7,7)
	3	Max Pooling	2×2,stride2	(64,4,4)
	4	Convolution+ReLU	64,3×3filters	(64,4,4)
	4	Upsampling	2×2,stride2	(64,8,8)
	5	Convolution+ReLU	32,3×3filters	(32,8,8)
Decoder	5	Upsampling	2×2,stride2	(32,16,16)
	6	Convolution+ReLU	16,3×3filters	(64,14,14)
	6	Upsampling	2×2,stride2	(64,28,28)
	7	Convolution+Sigmoid	1,3×3filters	(1,28,28)

Table 2. Architecture of DCAE on the SVHN and CIFAR-10 dataset

Stage	Layer	Layer Type	Size	Output Shape
	0	Input		(3,32,32)
	1	Convolution+ReLU	32,3×3filters	(32,32,32)
	1	Max Pooling	2×2,stride2	(32,16,16)
	2	Convolution+ReLU	64,3×3filters	(64,16,16)
	2	Max Pooling	2×2,stride2	(64,8,8)
	3	Convolution+ReLU	128,3×3filters	(128,8,8)
	3	Max Pooling	2×2,stride2	(128,4,4)
	4	Convolution+ReLU	256,3×3filters	(256,4,4)
	4	Max Pooling	2×2,stride2	(256,2,2)
	5	Convolution+ReLU	256,3×3filters	(256,2,2)
	5	Upsampling	2×2,stride2	(256,4,4)
	6	Convolution+ReLU	128,3×3filters	(128,4,4)
	6	Upsampling	2×2,stride2	(128,8,8)
	7	Convolution+ReLU	64,3×3filters	(64,8,8)
	7	Upsampling	2×2,stride2	(64,16,16)
	8	Convolution+ReLU	32,3×3filters	(32,16,16)
	8	Upsampling	2×2,stride2	(32,32,32)
	9	Convolution+Sigmoid	3,3×3filters	(3,32,32)



(a)

(b)

(c)

Fig. 5. Some original images (top lines) and reconstruction images (bottom lines) using DCAE on MNIST (a), SVHN (b) and CIFAR-10 (c) dataset.

3.2 Classification Performance Evaluation

In this section, we evaluate whether the features learned by our proposed method improve image classification performance as compared with other deep learning methods.

3.2.1 Comparison Method

We compare our proposed GAN-BDCAE with the following methods.

- Convolutional deep belief networks [37]: a traditional unsupervised feature learning algorithm with convolutional deep belief networks
- Stacked denoising autoencoders [20]: a traditional autoencoder method for unsupervised feature learning
- Deep NCAE [38]: a part-based representation learning machine with sparse autoencoders having nonnegativity constraints
- *k*-sparse autoencoder [39]: an autoencoder-based representation learning system that encourages sparsity

- Convolutional triangle *k*-means [40]: a method for selecting local receptive fields in deep networks
- Conv-WTA [41]: a winner-take-all method for learning sparse representations in an unsupervised fashion

3.2.2 Classification

In all of the experiments in this subsection, we compared the performance of GAN-BDCAE on the three benchmark datasets. We set *k* to 3 and used majority voting to aggregate the *k* DCAE outputs. We fine-tuned the DCAE filters with *N*-labeled data at the prediction stage.

First, we evaluated the effect of using different numbers of GAN-generated images during training. We expected the proposed method to learn more general knowledge as the number of unlabeled images increased. As shown in **Table 3**, the classification performance of our proposed method improved after adding some GAN-generated data to the training dataset. We obtained the best results when we added 50*k*, 80*k* and 50*k* generated inputs to the MNIST, SVHN, and CIFAR-10 datasets, respectively. Peak performance was activated when a nearly equal number of generated inputs were added to the real dataset. We observed improvements of 0.18% (from 0.87% to 0.69%), 1.42% (from 7.84% to 6.42%) and 2.14% (from 18.95% to 16.81%) on the three datasets, respectively.

Table 3. The GAN-BDCAE classification error (%) with #numbers of additionally generated data on the three datasets

# of generated data	0	1*k*	4*k*	10*k*	20*k*	50*k*	80*k*	100k
MNIST	0.87	0.86	0.86	0.77	0.76	**0.69**	0.71	0.72
SVHN	7.84	7.68	7.48	7.18	7.04	6.58	**6.42**	6.46
CIFAR-10	18.95	18.81	18.64	17.55	17.18	**16.81**	17.07	17.11

Next, we investigated the performance of GAN-BDCAE with *N* labeled real data. We first trained a GAN-BDCAE system on the new training dataset to learn the unsupervised features. We then trained the *k* softmax classifiers and fine-tuned the whole model by using *N* labeled samples. Finally, we compared the results with the traditional methods. **Tables 4-6** show the results: our GAN-BDCAE method offered the best performance on all three benchmark datasets. With fewer labeled data inputs (1*k* and 4*k*), our method also achieved a significant improvement in these image recognition tasks. To compare the performance of the traditional methods with and without GAN-generated data, we added 50*k*, 80*k* and 50*k* generated data to train these methods. We observed that the GAN-generated samples improved the performance of these traditional unsupervised methods on the image classification task.

Table 4. Test error (%) of GAN-BDCAE trained with *N* labeled samples on MNIST

Algorithm	*N*=1*k*	*N*=4*k*	ALL
Deep NCAE	-	-	~2.09
Stacked Denoising Autoencoders	-	-	1.28
Convolutional deep belief networks	-	-	0.82
k-sparse autoencoder	-	-	1.35
GAN- Deep NCAE	-	-	1.98
GAN- Stacked Denoising Autoencoders	-	-	1.11
GAN- Convolutional deep belief networks	-	-	0.77
GAN- *k*-sparse autoencoder	-	-	1.19
DCAE	2.36	1.76	0.99
BDCAE	2.22	1.74	0.87
GAN-BDCAE	**1.34**	**1.28**	**0.69**

Table 5. Test error (%) of GAN-BDCAE trained with *N* labeled samples on SVHN

Algorithm	*N*=1*k*	*N*=4*k*	ALL
Convolutional Triangle *k*-means	-	-	9.4
Conv-WTA	-	-	11.5
Stacked Conv-WTA	23.8	-	6.9
GAN- Conv-WTA	-	-	9.9
GAN- Stacked Conv-WTA	-	-	6.6
DCAE	29.1	17.5	7.5
BDCAE	24.2	16.2	7.3
GAN-BDCAE	**21.7**	**13.3**	**6.4**

Table 6. Test error (%) of GAN-BDCAE trained with *N* labeled samples on CIFAR-10

Algorithm	*N*=1*k*	*N*=4*k*	ALL
CDBN	-	-	21.1
Conv-WTA	-	-	19.9
Convolutional Triangle *k*-means	-	-	18.0
GAN- Conv-WTA	-	-	18.4
DCAE	47.7	32.2	19.8
BDCAE	45.1	30.4	18.5
GAN-BDCAE	**40.9**	**27.2**	**16.8**

4. Conclusion

In this paper, we presented a new unsupervised learning framework, namely GAN-BDCAE, for discriminative feature learning. This model can learn robust and discriminative feature representations from a mixture of real and generated data. We showed that the GAN-generated images effectively regularize BDCAEs during training. We mixed unlabeled GAN-generated images with real images for simultaneous semi-supervised learning. Albeit simple, the research results demonstrate consistent performance improvements over unsupervised learning methods, which offer support for the practical use of GAN-generated data. The BDCAEs method also shows its ability to learn robust features and improve the stability of single deep convolutional autoencoders. These facts reveal clear opportunities for designing more powerful representation learning by combining different improved techniques.

In the future, we will continue to investigate whether integrating GAN-generated images with better quality into unsupervised feature learning yields better performance for pattern recognition. We will also investigate the effects of BDCAEs on representation learning, including the influence of the architecture and the number of individuals.

References

[1] Noda, K., Yamaguchi, Y., Nakadai, K., Okuno, H. G., & Ogata, T., "Audio-visual speech recognition using deep learning," *Applied Intelligence*, 42(4), 722-737, 2015. Article (CrossRef Link)

[2] Wu, G., Lu, W., Gao, G., Zhao, C., & Liu, J., "Regional deep learning model for visual tracking," *Neurocomputing*, 175, 310-323, 2016. Article (CrossRef Link)

[3] LeCun, Y., Bengio, Y., & Hinton, G., "Deep learning," *nature*, 521(7553), 436, 2015. Article (CrossRef Link)

[4] Zhuang, F., Cheng, X., Luo, P., Pan, S. J., & He, Q., "Supervised Representation Learning with Double Encoding-Layer Autoencoder for Transfer Learning," *ACM Transactions on Intelligent Systems and Technology (TIST)*, 9(2), 16, 2017. Article (CrossRef Link)

[5] Feng, Z. H., Kittler, J., Awais, M., Huber, P., & Wu, X. J., "Wing Loss for Robust Facial Landmark Localisation With Convolutional Neural Networks," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2235-2245, 2018. Article (CrossRef Link)

[6] Hu, C., Wu, X. J., & Shu, Z. Q., "Discriminative Feature Learning via Sparse Autoencoders with Label Consistency Constraints," *Neural Processing Letters*, vol. 50(2), pp. 1079-1091, 2019. Article (CrossRef Link)

[7] Hu, C., Wu, X. J., & Kittler, J., "Semi-supervised learning based on GAN with mean and variance feature matching," *IEEE Transactions on Cognitive and Developmental Systems*, 2018. Article (CrossRef Link)

[8] Yang, J., Yu, K., Gong, Y., & Huang, T., "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. of Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 1794-1801, June 2009. Article (CrossRef Link)

[9] Lazebnik, S., Schmid, C., & Ponce, J, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. of 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, Vol. 2, pp. 2169-2178, 2006. Article (CrossRef Link)

[10] Li, H., Wei, Y., Li, L., & Chen, C. P., "Hierarchical feature extraction with local neural response for image recognition," *IEEE transactions on cybernetics*, 43(2), 412-424, 2013. Article (CrossRef Link)

[11] Zeiler, M. D., & Fergus, R. "Visualizing and understanding convolutional networks," in *Proc. of European conference on computer vision*, pp. 818-833, September 2014. Article (CrossRef Link)

[12] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 86(11), 2278-2324, 1998. Article (CrossRef Link)

[13] Schmidhuber, J., "Deep learning in neural networks: An overview," *Neural networks*, 61, 85-117, 2015. Article (CrossRef Link)

[14] Yuan, Y., Mou, L., & Lu, X., "Scene recognition by manifold regularized deep learning architecture," *IEEE transactions on neural networks and learning systems*, 26(10), 2222-2233, 2015. Article (CrossRef Link)

[15] Lu, X., Yuan, Y., & Yan, P., "Image super-resolution via double sparsity regularized manifold learning," *IEEE transactions on circuits and systems for video technology*, 23(12), 2022-2033, 2013. Article (CrossRef Link)

[16] Bourlard, H., & Kamp, Y., "Auto-association by multilayer perceptrons and singular value decomposition," *Biological cybernetics*, 59(4-5), 291-294, 1988. Article (CrossRef Link)

[17] Bengio, Y., "Learning deep architectures for AI," *Foundations and trends® in Machine Learning*, 2(1), 1-127, 2009. Article (CrossRef Link)

[18] Shin, H. C., Orton, M. R., Collins, D. J., Doran, S. J., & Leach, M. O., "Stacked autoencoders for unsupervised feature learning and multiple organ detection in a pilot study using 4D patient data," *IEEE transactions on pattern analysis and machine intelligence*, 35(8), 1930-1943, 2012. Article (CrossRef Link)

[19] Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A., "Extracting and composing robust features with denoising autoencoders," in *Proc. of the 25th international conference on Machine learning*, pp. 1096-1103, July 2008. Article (CrossRef Link)

[20] Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., & Manzagol, P. A., "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," *Journal of machine learning research*, 11(Dec), 3371-3408, 2010. Article (CrossRef Link)

[21] Masci, J., Meier, U., Cireşan, D., & Schmidhuber, J., "Stacked convolutional auto-encoders for hierarchical feature extraction," in *Proc. of International Conference on Artificial Neural Networks*, pp. 52-59, June 2011. Article (CrossRef Link)

[22] Radford, A., Metz, L., & Chintala, S., "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015. Article (CrossRef Link)

[23] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L., "Imagenet: A large-scale hierarchical image database," in *Proc. of Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248-255, June 2009. Article (CrossRef Link)

[24] Breiman, L., "Bagging predictors," *Machine learning*, 24(2), 123-140, 1996. Article (CrossRef Link)

[25] Ha, K., Cho, S., & MacLachlan, D., "Response models based on bagging neural networks," *Journal of Interactive Marketing*, 19(1), 17-30, 2005. Article (CrossRef Link)

[26] Mordelet, F., & Vert, J. P., "A bagging SVM to learn from positive and unlabeled examples," *Pattern Recognition Letters*, 37, 201-209, 2014. Article (CrossRef Link)

[27] Shu, Z., Zhao, C., & Huang, P., "Constrained Sparse Concept Coding algorithm with application to image representation," *KSII Transactions on Internet and Information Systems (TIIS)*, 8(9), 3211-3230, 2014. Article (CrossRef Link)

[28] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y., "Generative adversarial nets," in *Proc. of the Advances in neural information processing systems*, vol. 2, pp. 2672-2680, 2014. Article (CrossRef Link)

[29] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., & Chen, X., "Improved techniques for training gans," in *Proc. of the Advances in Neural Information Processing Systems*, pp. 2234-2242, 2016. Article (CrossRef Link)

[30] Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., & Abbeel, P., "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Proc. of the Advances in neural information processing systems*, pp. 2172-2180, 2016. Article (CrossRef Link)

[31] Pathak, D., Krahenbuhl, P., Donahue, J., Darrell, T., & Efros, A. A., "Context encoders: Feature learning by inpainting," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2536-2544, 2016. Article (CrossRef Link)

[32] Yeh, R., Chen, C., Lim, T. Y., Hasegawa-Johnson, M., & Do, M. N., "Semantic image inpainting with perceptual and contextual losses," *arXiv preprint, arXiv preprint arXiv:1607.07539*, 2, 2016.

[33] Hu, C., & Wu, X. J., "Autoencoders with Drop Strategy," in *Proc. of International Conference on Brain Inspired Cognitive Systems*, pp. 80-89, November 2016. Article (CrossRef Link)

[34] LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P., "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, 86(11), 2278-2324, 1998. Article (CrossRef Link)

[35] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., & Ng, A. Y., "Reading digits in natural images with unsupervised feature learning," in *Proc. of NIPS workshop on deep learning and unsupervised feature learning*, Vol. 2011, No. 2, p. 5, December 2011. Article (CrossRef Link)

[36] Krizhevsky, A., & Hinton, G., "Learning multiple layers of features from tiny images," *Technical report, University of Toro*nto, Vol. 1, No. 4, p. 7, 2009.

[37] Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y., "Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations," in *Proc. of the 26th annual international conference on machine learning*, pp. 609-616, June 2009. Article (CrossRef Link)

[38] Hosseini-Asl, E., Zurada, J. M., & Nasraoui, O., "Deep learning of part-based representation of data using sparse autoencoders with nonnegativity constraints," *IEEE transactions on neural networks and learning systems*, 27(12), 2486-2498, 2016. Article (CrossRef Link)

[39] Makhzani, A., & Frey, B., "K-sparse autoencoders," *arXiv preprint arXiv:1312.5663*, 2013. Article (CrossRef Link)

[40] Coates, A., & Ng, A. Y., "Selecting receptive fields in deep networks," in *Proc. of Advances in Neural Information Processing Systems*, pp. 2528-2536, 2011. Article (CrossRef Link)

[41] Makhzani, A., & Frey, B. J., "Winner-take-all autoencoders," *Advances in Neural Information Processing Systems*, pp. 2791-2799, 2015. Article (CrossRef Link)

Cong Hu received the B.Sc. degree from Jiangnan University, Wuxi, China, in 2009. He received the M.S. degree from Jilin University, Changchun, China in 2012. He is now a PhD candidate at the School of Internet of Things Engineering, Jiangnan University. He is now a visiting PhD student with Centre for Vision, Speech and Signal Processing, University of Surrey, UK, from 2018 to 2019. His current research interests include pattern recognition, computer vision, image processing, and deep learning.

Xiao-Jun Wu received the B.Sc. degree from Nanjing Normal University, Nanjing, China, in 1991. He received the M.S. degree and the Ph.D. degree in pattern recognition and intelligent systems from Nanjing University of Science and Technology, Nanjing, China, in 1996 and 2002, respectively. He is currently a Professor in artificial intelligent and pattern recognition at Jiangnan University. His current research interests include pattern recognition, computer vision, fuzzy systems, neural networks, and intelligent systems.

Zhen-Qiu Shu received the Ph.D. degree in Pattern Recognition and Intelligent Systems at Nanjing University of Science and Technology (NUST), China, in 2015. He was as a visiting student at Griffith University, Australia, from November 2014 to May 2015. He is currently an Associate Professor at Jiangsu University of Technology, Changzhou, China. His research interests mainly focus on machine learning and pattern recognition.