IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 27, 2019, accepted March 5, 2019, date of publication March 12, 2019, date of current version April 12, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2904614

Class-Incremental Learning Based on Feature Extraction of CNN With Optimized Softmax and One-Class Classifiers

XIN YE AND QIUYU ZHU
School of Communication and Information Engineering, Shanghai University, Shanghai 201900, China
Corresponding author: Qiuyu Zhu (zhuqiuyu@staff.shu.edu.cn)

ABSTRACT With the development of deep convolutional neural networks in recent years, the network structure has become more and more complicated and varied, and there are very good results in pattern recognition, image classification, scene classification, and target tracking. This end-to-end learning model relies on the initial large dataset. However, many data are gradually obtained in practical situations, which contradict the deep learning of one-time batch learning. There is an urgent need for an incremental learning approach that can continuously learn new knowledge from new data while retaining what has already been learned. This paper proposes an incremental learning algorithm based on convolutional neural network and support vector data description. CNN and AM-Softmax loss function are used to represent and continuously learn image features. Support vector data description is used to construct multiple hyperspheres for new and old classes of images. Class-incremental learning is achieved by the increment of hyperspheres. The experimental results show that the incremental learning method proposed in this paper can effectively extract the latent features of the image and adapt it to the learning situation of the class-increment. The recognition accuracy is close to batch learning.

INDEX TERMS One-class classifier, loss function, feature extraction, incremental learning.

I. INTRODUCTION

Natural vision systems are inherently incremental: new visual information is gradually incorporated while existing knowledge is preserved. For example, a child visiting the zoo will learn about many new animals without forgetting the pet at home. In contrast, most artificial object recognition systems can only be trained in a batch setting, where all object classes are known in advance and the training data of all classes can be accessed at the same time in arbitrary order. Incremental learning is an imitation of the cognitive process of human learning, and is possible to learn step by step without forgetting the knowledge that has already been learned. The concept of class-incremental learning is defined as follows:

1) it should be trainable from a stream of data in which examples of different classes occur at different times;
2) it should at any time provide a competitive multi-class classifier for the classes observed so far;
3) it can learn step by step without forgetting the knowledge that has already been learned.

The associate editor coordinating the review of this manuscript and approving it for publication was Bora Onat.

Interestingly, despite the vast progress that image classification has made over the last decades, there is not a satisfactory class-incremental learning algorithm nowadays. Although CNN has reached an unprecedented high level in the task of face recognition and image classification, it is based on the training of huge data sets and batch learning. For incremental learning, most classification tasks can only deal with a fixed number of categories or learn all kinds of images in one batch. Intuitively, one could try to overcome this by training classifiers from class-incremental data streams, e.g. using SGD optimization, however, this will cause a problem known as catastrophic forgetting in the literature [1].

In order to achieve class-incremental learning, we need to face two key issues: 1) how to extract effective features of images. 2) find a suitable incremental classification method to distinguish the new category from the old one, which has good generalization performance and excellent classification performance.

To solve these two problems, this paper combines convolutional neural network and one-class classifier SVDD to achieve incremental learning of classes. Undoubtedly,

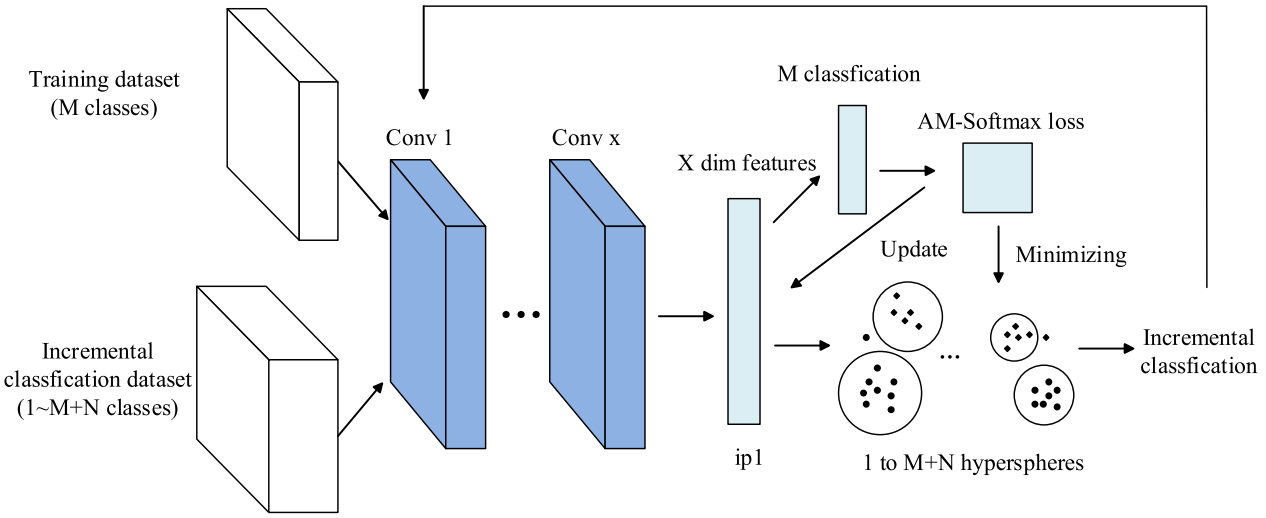

FIGURE 1. Overall system framework.

the convolutional neural network has proved its advantages in Feature Engineering in various tasks. Therefore, we use the convolutional neural network as a feature extraction tool and to learn and represent different types of samples. One-class classifier, SVDD, is often used for anomaly detection. In this paper, the class-incremental learning is innovatively divided into multiple incremental classifiers and the iterative updating is also very easy to understand. Moreover, SVDD encapsulates hyperspheres of high-dimensional data and combines them with AM-Softmax's feature processing method mentioned later, which is proven excellent incremental performance by experiments. The overall block diagram of our method is shown in Figure 1, whose details are given in Section 3.

Our main contributions are as follows:

1) We use a convolutional neural network combined with the improved loss function for feature extraction to obtain effective features with strong generalization characterization.

2) We propose a classifier design approach based on SVDD for class-incremental learning. The recognition rate curve of the verification set of the classifier determines the termination point of network training.

3) We evaluate our overall incremental learning approach in open datasets (MNIST, EMNIST, and CIFAR100), and experiments show that our method can effectively extract the latent features of images and implement the updating and iteration of class-incremental learning.

II. RELATED WORK

A. INTEGRATED LEARNING FOR INCREMENTAL LEARNING

Integrated learning [2] is to acquire multiple base classifiers through training samples and use a specific combination method to solve a problem together. Based on its characteristics, integrated learning is more suitable for incremental learning. According to the integration method, incremental learning can be divided into two categories: single classifier incremental learning and integrated incremental learning.

Single classifier incremental learning has only one learnable classifier from beginning to end, which can adjust the internal structure of the classifier according to the received sample information, to adapt to the new data. Although the single classifier incremental learning structure is relatively simple, in order to adjust the classifier structure and adapt to the new data, it is necessary to set the parameters manually. If these parameters are not properly selected, there will be problems with overfitting. In the process of incremental learning, the internal structure of the single classifier needs to be continuously adjusted, and it is difficult to estimate the storage space, to predict speed, etc., so that we can't choose the suitable algorithm to train the data, and structural adjustments can also easily lead to catastrophic forgetting of learned knowledge.

Related algorithms include ARTMAP [3] (Adaptive Resonance Theory modules map), EFuNNs [4] (Evolving Fuzzy Neural Networks), incremental decision trees [5], incremental SVM [6] and so on.

Integrated incremental learning is to train each new data into a new model, and then combine these models for comprehensive utilization. The integrated model thus contains incremental information each time to enable incremental learning. Compared with single classifier incremental learning, it is more robust.

In 2001, the Learn++ algorithm was proposed by Polikar *et al.* [7], which is a supervised incremental learning based on AdaBoost. The weight can be given to the new data, and then the update of sample weight is performed according to the result of the classification, and the weak classifier trained with the new data is added to the integrated classifier. In 2003, Seipone and Bullinaria [8] proposed EEN (Evolved Neural Network) based on traditional neural network. Facing the changes of the data sample environment, genetic algorithms were used to evolve the connection weights, network

layers, learning rates and other parameters of the neural network. When there is new data, the network structure can be changed to achieve incremental learning. In 2005, Tnoue and Narihisa [9] proposed SONG (Self-Organizing Neural Grove) as an integrated incremental learning based on self-generated neural trees. All training sets are constructed into trees, and training samples correspond to the leaf nodes of trees. Through the multiple input of training samples in different orders, multiple generated trees are obtained to construct the integrated classifier, and finally the incremental learning is completed by using the pruning algorithm to reduce the time and space overhead.

B. INCREMENTAL LEARNING WITH DEEP LEARNING

Xiao *et al.* [10] proposed a network that can grow hierarchically. Each node is composed of clusters of similar classes. Through the tree structure, only the local part of the model needs to be adjusted when the model is updated, and the adjustment can be strictly controlled. Incremental learning is realized through the growth of the network, but the solution faces the difficulty of training the large network and how to effectively increase the network capacity. Aiming at the catastrophic forgetting problem in incremental learning of convolutional neural networks, Rusu and Andrei [11] proposed a progressive NN to solve the problem of adapting the network to new tasks (incremental). The idea is to keep all the networks of the previous tasks, create a new network for each new task, and retain the low-level features of the old network. This method can better solve the problem of catastrophic forgetting, but it comes with the continuously growth of the network scale, and the need for manual knowledge in the design of different tasks.

Venkatesan and Ragav [12] uses GAN to generate Phantom Sampling to retain the information of the original training samples. These phantom samples are used to train new deep networks together with incremental samples, achieving better class incremental training effects. However, this method takes a long time to train and is difficult to apply to the case of new incremental samples of old categories. ICaRL proposed by Rebuffi *et al.* [13] uses convolutional neural network for feature learning and characterization. The new class samples and the previously stored old class samples are jointly trained in convolutional neural network to update the current model parameters and obtain new feature representations. In the classification, the NCM [14] idea is used to classify the extracted feature vectors in the sample set by using Nearest-Mean-of-Exemplars. Based on iCaRL, Wu and Yue [15] redefined the loss function (cross-entropy loss function + distillation loss function), and added GANS [16] to generate a few samples of the old categories to improve generalization ability.

III. METHOD

A. FEATURE REPRESENTATION

For the incremental classification, whether it is new class or old class, it is very important to characterize the image or to continuously learn the image features. In the past, the improvement of network mostly focused on the design of network structure, which is deeper and more complicated, but the most critical impact on feature distribution is the objective function or loss function. This paper introduces the AM-Softmax [17] function to improve the intra-class and inter-class distance obviously, to achieve better incremental classification.

Let's review the traditional Softmax loss first. We define the i-th input feature $\boldsymbol{X}_i$ with the label y_i. Then the Softmax loss can be written as

$$L = \frac{1}{N}\sum_i L_i = \frac{1}{N}\sum_i -log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right) \tag{1}$$

where f_j denotes the j-th element of the vector of class scores $\boldsymbol{f}$, and N is the number of training data. In the Softmax loss, $\boldsymbol{f}$ is usually the activations of a fully connected layer $\boldsymbol{W}$, so f_{y_i} can be written as $f_{y_i} = \boldsymbol{W}_{y_i}^T\boldsymbol{X}_i$, Thus the loss becomes

$$L_i = -log\left(\frac{e^{\|\boldsymbol{W}_{yi}\|\|\boldsymbol{X}_i\|cos(\theta_{y_i})}}{\sum_j e^{\|\boldsymbol{W}_j\|\|\boldsymbol{X}_i\|cos(\theta_j)}}\right) \tag{2}$$

where $0 \le \theta_j \le \pi$.

Before AM-Softmax, L-Softmax proposed by Liu *et al.* [18] introduced the concept of angular margin to add a parameter m to change the cos distance of the weights $\boldsymbol{W}$ and $\boldsymbol{X}$ to $cos\,(m\theta)$, and adjust the distance between features by m. Although L-Softmax enlarges the learning difficulty of loss function through parameter m and can significantly improve the intra-class and inter-class distance, the learning difficulty of $cos\,(m\theta)$ increases exponentially when the number of classes increases dramatically. Inspired by these methods, the author proposes a more intuitive and easy-to-understand method-Additive Margin Softmax (AM-Softmax). Similar to the former, AM-Softmax rewrites the expression of $cos\,(\theta)$ as $cos\,(\theta) - m$.

The above formula is simpler than L-Softmax in form and calculation. In addition, on the basis of L-softmax, the weight and the feature vector are normalized: $b = 0$, $\|\boldsymbol{W}\| = 1$, $\|\boldsymbol{X}\| = 1$. Compared with L-Softmax loss, the difference between the classes is only related to the angle θ. In the three-dimensional feature space, we can see that all classes of features are distributed on the sphere. Visualizing MNIST dataset as shown in Figure 2, we can clearly see that AM-Softmax has a strong constraint on feature expression. This distribution is also easier to integrate with subsequent SVDD.

To sum up, we can write AM-Softmax loss function as:

$$L_{AM} = -\frac{1}{N}\sum_i log\frac{e^{s\cdot(cos\theta_{y_i}-m)}}{e^{s\cdot(cos\theta_{y_i}-m)} + \sum_{j=1,j\neq y_i}^{c} e^{s\cdot cos\theta_j}} \tag{3}$$

At the same time, the scale factor s is added to control the scaling. In this paper, the fixed value 10 is used to accelerate the convergence and make it more stable.

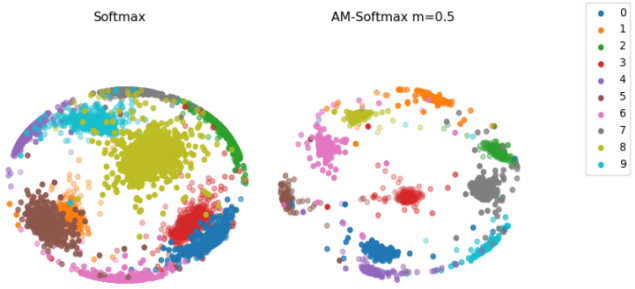

FIGURE 2. AM-Softmax feature distribution.

B. SVDD

SVDD is a classification method based on Support Vector Machine (SVM) [19] proposed by TAX [20] where a hypersphere is used to separate the data instead of a hyperplane. The main algorithm idea is to map data samples $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_N\}$ to high-dimensional feature spaces through $\phi(\boldsymbol{x}_i)$. The objective of SVDD is to find the smallest hypersphere with center $\mathbf{a}$ and radius $R > 0$ that encloses the majority of the data in feature space. The SVDD primal problem is given by

$$\min F(R, \mathbf{a}) = R^2 + C \sum_{i=1}^{N} \xi_i \tag{4}$$

constraint condition is

$$s.t.\ \|\phi(\boldsymbol{x}_i) - \mathbf{a}\|^2 \leq R^2 + \xi_i, \quad \xi_i \geq 0,\ i = 1, 2, \ldots, N \tag{5}$$

where N is the total number of samples; the distance from any point $\boldsymbol{x}_i$ to the center of the hypersphere $\mathbf{a}$ is $\|\phi(\boldsymbol{x}_i) - \mathbf{a}\|$; R is the radius of the hypersphere; ξ_i is the slack variable; the parameter C controls the trade-off between the volume and the errors.

Constraints (5) can be incorporated into Eq. (4) by using Lagrange multipliers:

$$\begin{aligned} L(R, \mathbf{a}, a_i, \gamma_i, \xi_i) = {} & R^2 + C \sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \gamma_i \xi_i \\ & - \sum_{i=1}^{N} a_i (R^2 + \xi_i - \|\phi(\boldsymbol{x}_i) - \mathbf{a}\|^2) \end{aligned} \tag{6}$$

where the Lagrange multipliers $a_i \geq 0$ and $\gamma_i \geq 0$. For each $R, \mathbf{a}, a_i, \xi_i$ the partial derivatives are derived and make it equal to 0, namely:

$$\begin{cases} \dfrac{\partial L}{\partial R} = 0 \Rightarrow \displaystyle\sum_{i=1}^{N} a_i = 1 \\ \dfrac{\partial L}{\partial \mathbf{a}} = 0 \Rightarrow \mathbf{a} = \dfrac{\sum_i a_i \phi(\boldsymbol{x}_i)}{\sum_i a_i} = \displaystyle\sum_{i=1}^{N} a_i \phi(\boldsymbol{x}_i) \\ \dfrac{\partial L}{\partial \xi_i} = 0 \Rightarrow C - a_i - \gamma_i = 0 \end{cases} \tag{7}$$

substituting (7) into (6), we can obtain the following equation:

$$L = \sum_{i=1}^{N} a_i (\phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_i)) - \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j (\phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_j)) \tag{8}$$

Equation (8) is a standard quadratic optimization problem with an optimal solution a_i. In the actual calculation, $a_i = 0$ is the majority, and a few x_i with $a_i > 0$ are the support vector.

To test an object $\boldsymbol{Z}$, the distance to the center of the sphere must be calculated. A test object $\boldsymbol{Z}$ is accepted when this distance is smaller or equal than the radius:

$$\begin{aligned} \|\phi(\mathbf{Z}) - \mathbf{a}\|^2 = {} & \phi(\mathbf{Z}) \cdot \phi(\mathbf{Z}) - 2 \sum_{i=1}^{N} a_i \phi(\boldsymbol{x}_i) \cdot \phi(\mathbf{Z}) \\ & + \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j (\phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_j)) \end{aligned} \tag{9}$$

By definition, R^2 is the distance from the center of the sphere $\mathbf{a}$ to the boundary. Support vectors which fall outside the description ($a_i = C$) are excluded. Therefore:

$$\begin{aligned} R^2 = {} & \phi(\boldsymbol{x}_k) \cdot \phi(\boldsymbol{x}_k) - 2 \sum_{i=1}^{N} a_i \phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_k) \\ & + \sum_{i=1}^{N} \sum_{j=1}^{N} a_i a_j (\phi(\boldsymbol{x}_i) \cdot \phi(\boldsymbol{x}_j)) \end{aligned} \tag{10}$$

for any $\boldsymbol{x}_k \in SV_{<C}, a_k < C$.

C. CLASS-INCREMENTAL CLASSIFICATION

This section describes the system framework and how to achieve class-incremental classification. Firstly, the convolution neural network is used to extract and represent features. In order to obtain more compact features within classes and more distinctive features between classes, we adopt the latest AM-Softmax loss function to update and iterate the parameters of CNN, so that the image features obtained by the non-linear mapping of the CNN can meet the training of SVDD.

Secondly, the feature vectors of the class incremental dataset are obtained by forward propagation in the linear layer ip1. We use SVDD to train the samples, wrap the samples with a hypersphere. With the iteration of training, the intra-class feature is more compact and the volume of hypersphere is reduced, which makes the expression ability strong and the boundary between classes obvious, so that the robustness and classification performance of the subsequent incremental classification are better. The support vectors and the radius of the sphere center of each hypersphere in the feature space can be preserved.

1) AM-SOFTMAX AND SVDD

As can be seen from Figure 2, AM-Softmax improves the feature distribution significantly. Cross Entropy loss and

Softmax functions are enough to solve many problems in the original closed-set image classification. However, in an open-set classification task (such as face recognition), in a specific metric space, the maximum distance within the same class is required to be smaller than the minimum distance between different classes. In the traditional Euclidean distance metric space, the data distribution is scattered when the number of classes increases, resulting in performance degradation.

AM-Softmax adds restrictions to normalize weights and feature vectors in classification layers: $bias = 0, \|\boldsymbol{W}\| = 1, \|\boldsymbol{X}\| = 1$. The bias term is set to 0, mainly for the convenience of geometric analysis, and many experiments show that the bias term has little effect on the results. By weight normalization, the impact of imbalanced training data is reduced. The feature vectors normalization can reduce the impact of excessive data scene differences. Finally, the metric space is converted to the hypersphere, and the margin is added to increase the learning difficulty and obtain a better feature distribution. Therefore, this metric space distribution coincides with the idea of SVDD.

By encapsulating intra-class data in hyperspheres and increasing the distance between classes, incremental learning is divided into one-class classification by increasing the number of hyperspheres, which realizes the training on open sets and the possibility of incremental classification.

2) REJECTION RATE

Finally, a definition of rejection rate is introduced to adapt to incremental learning scenarios. For hypersphere, in order to reduce the volume of the sphere and redundancy space, only all training samples are included as far as possible. Then in the testing process, there will be some samples that do not belong to any hypersphere in the current metric space, so these samples are rejection samples, and thus there is rejection rate for each class. After all, for pattern classification, the cost of incorrect samples is often much higher. Rejected samples can be retained and subsequent classified (such as updating hypersphere support vectors).

The definitions of rejection rate are: let test class $\boldsymbol{T} = \{\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_N\}$, there are n hyperspheres in metric space, *i.e.* n classes, center are $\{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n\}$, radius are $\{R_1, R_2, \ldots, R_n\}$. Rejected samples $\{\boldsymbol{t}_1, \boldsymbol{t}_2, \ldots, \boldsymbol{t}_k\}$ satisfy

$$\left\|\phi(\boldsymbol{t}_i) - \boldsymbol{a}_j\right\|^2 \leq R_j^2 + \xi, \quad i = 1, 2, \ldots, k. \, j = 1, 2, \ldots, n \tag{11}$$

where ξ is the slack variable of this kind of training set. Thus, the rejection rate is:

$$\boldsymbol{T} = \frac{k}{N} \tag{12}$$

IV. EXPERIMENTS AND RESULTS

A. CNN ARCHITECURES

This paper uses WideResNet [22] as the basic CNN network. ResNet's [23] has an indelible place in deep learning. WideResNet widens the Basic Block in ResNet, and increases the number of channels, to improve the accuracy, reduce the number of network layers, and speed up network training. The following experimental network structure is shown in Table 1.

TABLE 1. CNN architectures.

Group name	Block type=$B(3,3)$
Conv1	$[3 \times 3, 16]$
Conv2	$\begin{bmatrix} 3 \times 3, 16 \times 10 \\ 3 \times 3, 16 \times 10 \end{bmatrix} \times 4$
Conv3	$\begin{bmatrix} 3 \times 3, 32 \times 10 \\ 3 \times 3, 32 \times 10 \end{bmatrix} \times 4$
Conv4	$\begin{bmatrix} 3 \times 3, 64 \times 10 \\ 3 \times 3, 64 \times 10 \end{bmatrix} \times 4$
Avg-pool	$[7 \times 7]/[8 \times 8]$
Fully Connected ip1	100
Fully Connected ip2	Softmax

TABLE 2. The penalty coefficient C of SVDD (AC = ACCURACY).

C	CNN+AM-Softmax+SVDD		
	Ac	Error	Reject
0.01	95.47	0.46	4.07
0.1	**97.84**	1.01	1.15
0.5	97.31	2.16	0.53
1	97.02	2.76	0.22

Among them, the convolution layer unified convolution kernel size is 3×3, and padding is 1. All network structures and training were implemented on Pytorch with an initial learning rate of 0.1, a weight decay of 0.0005, and a momentum of 0.9. The first fully connected layer output dimension is fixed at 100 dimensions for feature output.

The system framework is shown in Figure 1. The two input data sets do not interfere with each other. The M-classes data sets are used to train the network, and the AM-Softmax loss function is used to optimize parameters and nonlinear mapping. The features of training dataset are extracted in ip1 layer by the network. The trained SVDD in every epoch output the incremental classification result, and the AM-Softmax reduces the hypersphere volume to increase incremental classification result. Finally, the training of network and SVDD is stopped when the classification performance of the verification set in the incremental data set reaches the best level.

Firstly, the MNIST data set is initially classified for different penalty coefficient C of SVDD. The value of C in SVDD ranges from 0-1. The experimental results are as follows:

We can see that different C will affect the classification results. Intuitively understanding, the bigger C is, the bigger the hypersphere is, and the more attention is paid to the edge samples. The smaller C is, the smaller the sphere is, and the discarding part of the edge samples makes the accuracy and size compatible.

The penalty coefficient C of SVDD is set to 0.1 in the next experiment. For each class of hyperspheres, support vectors, sphere center and radius are preserved, and training samples are discarded. When new untrained samples are added, feature representation is used to construct hyperspheres until the training is completed.

In the next incremental classification experiment, there are two types of data sets, MNIST/EMNIST and CIFIR100. In contrast experiment, MNIST/EMNIST data sets, because of its simplicity, we use the traditional Kernel-based SVDD for comparison, which normalize the original image, perform RBF mapping, and achieve incremental classification. In CIFAR100 data set, due to too high dimension, Kernel-based SVDD cannot be processed well, thus only two kinds of Softmax+SVDD are compared.

B. MNIST/EMNIST

MNIST [23] is a handwritten digital dataset and is widely used as an introductory training set for convolutional neural networks. The EMNIST [24] dataset is derived from the National Institute of Standards and Technology (NIST). It is made up of 810,000 characters images handwritten by 3,600 people. The dataset includes a total of 62 classes of Arabic numerals ''0∼9'', lowercase English letters ''a∼z'', and uppercase English letters ''A∼Z''. In order to match with MNIST, it is uniformly converted into a 28 × 28 grayscale picture format.

In this section, the number of training dataset classes is M = 30, and the incremental test classification datasets is increased from 10 to 40 classes, in which $N = 40 - 30 = 10$ is the incremental classes. The training set, verification set and test set have 1500 images, 500 images and 500 images respectively.

The network is only trained for 30 classes, and gives multi-classification results. The incremental data set has 40 classes, 10 classes are untrained data sets, and the image features are uniformly output with the full connection layer ip1, which uses SVDD to achieve compatibility and incremental classification of new classes and old classes. The experimental results are shown in Table 3.

In this experiment, we can see that the incremental classification by combining SVDD with CNN is greatly improved compared with the traditional SVDD classification, and AM-Softmax also brings about the improvement of recognition accuracy, which shows that the feature distribution of the trained CNN is more reasonable and more suitable for the incremental classification of SVDD. Moreover, our method can also participate in the recognition of new classes which have not been trained, and is compatible with the old classes. Each hypersphere does not interfere with each other, and must be opened in the metric space as far as possible. Compared with traditional CNN, although the accuracy of SVDD in fixed category recognition is slightly lower, the error rate is close, and our method can incrementally classify new categories. Among them, the average R^2 of hyperspheres is about 600, 180 and 100 respectively.

TABLE 3. MNIST/EMNIST incremental experiment results(Ac=Accuracy).

%	Kernel-SVDD			CNN+Softmax+SVDD		
	Ac	Error	Reject	Ac	Error	Reject
10	74.62	23.25	2.13	97.23	1.55	1.22
15	71.34	26.79	1.87	96.57	2.4	1.03
20	68.41	30.04	1.55	96.14	3.07	0.79
25	65.27	33.49	1.24	95.47	3.97	0.56
30	64.03	34.99	0.98	95.06	4.46	0.48
35	62.13	37.3	0.57	93.89	5.79	0.32
40	60.28	39.29	0.43	92.61	7.18	0.21

%	CNN+AM-Softmax+SVDD			CNN+AM-Softmax	
	Ac	Error	Reject	Ac	Error
10	**97.84**	1.01	1.15	99.02	0.98
15	**97.28**	1.76	0.96	98.88	1.12
20	**97.03**	2.26	0.71	98.99	1.01
25	**96.35**	3.17	0.48	98.96	1.04
30	**96.04**	3.63	0.33	98.97	1.03
35	**94.75**	4.99	0.26	/	/
40	**93.48**	6.35	0.17	/	/

In the SVDD incremental classification, there is a value of the rejection rate, meaning that part of the test samples fall outside all the hyperspheres. From another point of view, this is also the advantage of the hypersphere incremental classification. Keeping some samples that are not correctly recognized can take additional measures instead of directly accounting for errors. AM-Softmax also performs best in incremental classification, which also has a strong representation performance, and the recognition rate is close to the end-to-end classification effect of the batch convolutional neural network.

C. CIFAR100

The CIFAR100 dataset has a total of 100 classes of data, a training set of 500 images per class, and a test set of 100 images per class. As above, in this section, M = 80, N = 20, and the incremental classification data set is increased from 10 to 100 classes. Each class of verification set draws 100 images from the training set.

In the training phase, we follow the standard data augmentation [25] for training: 4 pixels are padded on each side, and a 32 × 32 crop is randomly sampled from the padded image or its horizontal flip. In the testing phase, we only evaluate the single view of the original 32 × 32 image. The incremental data set does not make any changes to the image. The incremental classification results are shown in Table 4.

TABLE 4. CIFAR100 incremental experiment results (Ac = Accuracy).

%	CNN+Softmax+SVDD			CNN+AM-Softmax+SVDD			CNN+AM-Softmax	
	Ac	Error	Reject	Ac	Error	Reject	Ac	Error
10	81.35	2.38	16.27	**85.36**	1.39	13.25	77.52	22.48
30	75.52	10.52	13.96	**80.13**	11.56	8.31	76.57	23.43
50	73.41	16.24	10.35	**77.92**	15.7	6.38	77.89	22.11
80	69.91	26.62	3.47	**73.15**	24.29	2.56	76.65	23.35
90	66.35	32.3	1.35	**69.46**	29.32	1.22	/	/
100	63.03	36.45	0.52	**67.98**	31.78	0.24	/	/

The performance of applying SVDD to incremental classification of 100-classes is excellent, which also shows that our CNN + AM-Softmax non-linear mapping can be well combined with SVDD. Compared with traditional CNN classification, SVDD can have a smaller error rate and a higher accuracy in fewer classes, because hyperspheres can have penalty coefficients to control the size of spheres, and can appropriately expand the volume of spheres in metric spaces with fewer classes. The average R^2 of hyperspheres is about 90 and 40 respectively.

From the two experimental results, our method is shown to be more capable of expressing features in class-incremental learning scenarios, which intuitively enhances the intra-class compactness and inter-class separability. Moreover, using SVDD for class increment classification also achieves better performance. In the case of an increase in the number of classes, the overall average accuracy of MNIST/EMNIST can be maintained at a relatively high level, and the performance in the CIFAR100 is also at a good level of recognition rate.

V. CONCLUDING REMARKS

In our work, we propose a new classification method for class-incremental learning. We trained the convolutional network with the improved Softmax loss function as a feature extraction network to perform more efficient and generalized feature representations on images, and then use one-class classifier SVDD to implement incremental classification. In the case of an increase in the number of classes, the original hypersphere model is retained, and the new hypersphere is iteratively updated to ensure that the overall classification network performance maintains a certain accuracy rate.

Our future research directions include: (1) Base on the support vector of hypersphere of SVDD, SVM classifier can be used to achieve classification of positive and negative support vector samples, to improve the recognition rate of class incremental learning, and provide solution of the rejection samples. (2) To combine SVDD with clustering to achieve the split of the hypersphere and the increment of the classes, (3) To optimize the SVDD construction process, such as making use of different nonlinearly kernel function, and multiple hyperspheres for one class, etc.

REFERENCES

[1] M. McCloskey and N. J. Cohen, "Catastrophic interference in connectionist networks: The sequential learning problem," *Psychol. Learn. Motivat.*, vol. 24, pp. 109–165, Dec. 1989.
[2] G. Valentini and F. Masulli, "Ensembles of learning machines," in *Italian Workshop Neural Nets-Revised Papers*. Springer-Verlag, 2002, pp. 3–22.
[3] D. Silver *et al.*, "Mastering the game of Go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.
[4] N. Kasabov, "Evolving fuzzy neural networks for supervised/unsupervised online knowledge-based learning," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 31, no. 6, pp. 902–918, Dec. 2001.
[5] P. E. Utgoff, "Incremental induction of decision trees," *Mach. Learn.*, vol. 4, no. 2, pp. 161–186, 1989.
[6] S. Ruping, "Incremental learning with support vector machines," in *Proc. IEEE Int. Conf. Data Mining*, Nov./Dec. 2001, pp. 641–642.
[7] R. Polikar, L. Upda, and S. S. Upda, and V. Honavar, "Learn++: An incremental learning algorithm for supervised neural networks," *IEEE Trans. Syst., Man, Cybern. C, Appl. Rev.*, vol. 31, no. 4, pp. 497–508, Nov. 2001.
[8] T. Seipone and J. A. Bullinaria, "Evolving improved incremental learning schemes for neural network systems," in *Proc. IEEE Congr. Evol. Comput.*, vol. 3, Sep. 2005, pp. 2002–2009.
[9] H. Inoue and H. Narihisa, "Self-organizing neural grove and its applications," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, vol. 2, Jul./Aug. 2005, pp. 1205–1210.
[10] T. Xiao, J. Zhang, K. Yang, Y. Peng, and Z. Zhang, "Error-driven incremental learning in deep convolutional neural network for large-scale image classification," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 177–186.
[11] A. A. Rusu *et al.* (Jun. 15, 2016). "Progressive neural networks." [Online]. Available: https://arxiv.org/abs/1606.04671
[12] R. Venkatesan, H. Venkateswara, S. Panchanathan, and B. Li. (May 2, 2017). "A strategy for an uncompromising incremental learner." [Online]. Available: https://arxiv.org/abs/1705.00744
[13] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 5533–5542.
[14] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka, "Metric learning for large scale image classification: Generalizing to new classes at near-zero cost," in *Proc. Eur. Conf. Compu. Vis.*, 2012, pp. 488–501.
[15] Y. Wu *et al.* (Feb. 2, 2018). "Incremental classifier learning with generative adversarial networks." [Online]. Available: https://arxiv.org/abs/1802.00853
[16] M. Mirza and S. Osindero, "Conditional generative adversarial nets," in *Proc. Deep Learn. Workshop NIPS*, 2014.
[17] F. Wang, W. Liu, H. Liu, and J. Cheng, "Additive margin softmax for face verification," *IEEE Signal Process. Lett.*, vol. 25, no. 7, pp. 926–930, Jul. 2018.
[18] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," in *Proc. 33rd Int. Conf. Mach. Learn.*, 2016, pp. 507–516.
[19] L. Bottou and C.-J. Lin, "Support vector machine solvers," in *Large Scale Kernel Machines*, L. Bottou, O. Chapelle, D. DeCoste and J. Weston, Eds. Cambridge, MA, USA: MIT Press, 2007.
[20] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, 2004.
[21] S. Zagoruyko and N. Komodakis. (May 23, 2016). *Wide Residual Networks*. [Online]. Available: https://arxiv.org/abs/1605.07146
[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Compu. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[23] Y. Lécun and C. Corninna. *The Minist Database of Handwritten Digits [DB/OL]*. [Online]. Available: http://yann.lecun.com/exdb/mnist/
[24] G. Cohen, S. Afshar, J. Tapson, and A. van Schaik. (Feb. 17, 2017). "EMNIST: An Extension of MNIST to handwritten letters." [Online]. Available: https://arxiv.org/abs/1702.05373
[25] C.-Y. Lee, S. Xie, P. W. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Proc. 18th Int. Conf. Artif. Intell. Statist.*, 2015, pp. 562–570.

XIN YE received the bachelor's degree in engineering from the Information Engineering College, Zhejiang University of Technology, in 2012. He is currently pursuing the degree with the School of Communication, Shanghai University. His research interests include computer vision and pattern recognition.

He received the National First Prize in the Electronic Design Competition during his undergraduate course. During the postgraduate study, he participated in projects such as face recognition and published a number of EI conference papers. He is mainly engaged in the study of incremental learning.

QIUYU ZHU received the B.Sc. degree from the Department of Electronic Engineering, Fudan University, in 1985, and the master's degree in engineering from the Department of Electronic Engineering, Shanghai University of Science and Technology, in 1988, and the Ph.D. degree in information and communication engineering from the School of Communication, Shanghai University, in 2006.

Since 1988, he has been a Teacher, where he has been teaching and researching in image processing, pattern recognition, smart city, and computer application. As a project leader or a backbone, he has undertaken more than ten vertical scientific research projects, including the National Natural Science Foundation, the National 863 Special Project, the Shanghai Municipal Science and Technology Commission, and the Shanghai Municipal Education Commission, including six major scientific and technological projects under the Shanghai Science and Technology Commission. He has undertaken more than 20 horizontal scientific research projects, among which the container code recognition, binocular passenger flow counter, and automatic vehicle detection systems, which have been researched and developed, have been scaled up. At present, more than 60 research papers have been published, including more than 20 papers in three major research papers.