Progressive Class-based Expansion Learning For Image Classification

Hui Wang, Hanbin Zhao, and Xi Li

Abstract—In this paper, we propose a novel image process scheme called class-based expansion learning for image classification, which aims at improving the supervision-stimulation frequency for the samples of the confusing classes. Class-based expansion learning takes a bottom-up growing strategy in a class-based expansion optimization fashion, which pays more attention to the quality of learning the fine-grained classification boundaries for the preferentially selected classes. Besides, we develop a class confusion criterion to select the confusing class preferentially for training. In this way, the classification boundaries of the confusing classes are frequently stimulated, resulting in a fine-grained form. Experimental results demonstrate the effectiveness of the proposed scheme on several benchmarks.

Index Terms—Class-based expansion optimization, image classification.

I. INTRODUCTION

CONVOLUTIONAL neural networks [1]–[5] (CNN) have attracted considerable attention in image classification due to their effectiveness in representation learning [6], [7]. Since they require computationally expensive and memory-consuming operations, CNN training typically resorts to stochastic gradient descent (SGD) [8], [9] for iterative batch-level learning, it traverses the entire training dataset across randomly generated batches throughout successive epochs. With this epoch-by-epoch learning procedure, the classification boundaries of the CNN model are dynamically updated until convergence. Due to the memory resource limit, the samples within a smaller-size batch usually distribute extremely diversely and sparsely, resulting in a low supervision-stimulation frequency for each sample. The low frequency for each sample in turn causes the learning process to pay more attention to the learning quality of the coarse-grained classification boundaries while ignoring fine-grained details. Therefore, seeking for an effective and stable image classification strategy remains a key issue to solve in the CNN learning area.

To date, curriculum learning [10]–[15] and self-paced learning [16]–[22] have been proposed to improve the speed of convergence of the training process to a minimum and the quality of the local minima obtained. The key concept of the proposed methods is inspired by human behavior, who always learn new things from “easy” to “complex”. But these methods still do not consider the fine-grained classification boundaries.

In this letter, we propose a new learning pattern that arises from the inspirations of the biological learning mechanism. The Hebbian theory [23] delivers an important insight that the increase in synaptic effects of synaptic cells comes from repeated and sustained stimulation. Meanwhile, the human learning pattern usually follows a progressive knowledge expansion learning pipeline, it dynamically learns new knowledge while keeping the old knowledge frequently reviewed. The knowledge that is frequently reviewed is often better learned. Inspired by the biological learning mechanism, we propose a progressive learning pipeline that aims at effectively enhancing the supervision-stimulation frequency for each sample to enhance the quality of the fine-grained classification boundaries as shown in Fig. 1. Specifically, we present a progressive piecewise class-based expansion learning scheme, which first learns fine-grained classification boundaries for a small portion of classes and subsequently expands the classification boundaries with new classes added. Therefore, the presented class-based expansion learning scheme takes a bottom-up growing strategy in a class-based expansion optimization fashion, which puts more emphasis on the quality of learning the fine-grained classification boundaries for dynamically growing local classes. Besides, we propose a class confusion criterion to sort the classes involved in the class-based expansion learning process. The classes where the samples have large intra-class and small inter-class distances on average (i.e., class confusing samples) are preferentially involved in the class-based expansion learning process. Once a particular class is selected, all the samples belonging to this class are added to the training sample pool for the CNN model learning. Such an expansion procedure is repeated until all the class samples participated in the training process. In this way, the classification network model is dynamically refined based on the updated training sample pool, and the classification boundaries of the preferentially selected classes are frequently stimulated, resulting in a fine-grained form.

The main contributions of this work are summarized as...
follows: i) Motivated by Hebbian theory, we propose to investigate the influence of “stimulation frequency” on neural network learning and make an observation that the poor performance of the confusing classes is partially a result of the low stimulation frequency. ii) We propose a novel class-based expansion learning pipeline to deal with the learning problem. This pipeline progressively trains the CNN model in a hard-to-easy class-based growing manner, thereby the classification boundaries of the preferentially selected confusing classes are frequently stimulated. iii) We develop two class confusion criteria to sort the classes for the class-based expansion learning process. Extensive experiments demonstrate the effectiveness of our work against conventional learning pipelines on several benchmarks.

II. METHOD

In this section, we detail the proposed class-based expansion learning scheme. We first formally define the problem in Section II-A, we describe our algorithm to solve it in Sections II-B and II-C.

A. Problem Definition

Given a $M$-classes dataset $D = \{C_1, C_2, \ldots, C_M\}$, and the $m$-th class of the dataset $C_m$ contains $N_m$ samples $x^m$ and their corresponding labels $y^m$:

$$C_m = \{(x_1^m, y_1^m), (x_2^m, y_2^m), \ldots, (x_{N_m}^m, y_{N_m}^m)\}. \quad (1)$$

Let $f(\cdot; \theta)$ denote the mapping function of the CNN model, where $\theta$ represents the model parameters inside $f(\cdot)$. In a typical training process, the goal is to learn an optimal $\theta^*$:

$$\theta^* = \text{arg min } \sum_{(x, y) \in D} l(f(x; \theta), y), \quad (2)$$

where $l(\cdot, \cdot)$ is the loss function (e.g. cross-entropy loss).

B. Class Confusion Criterion

In this section, we introduce our metric for deciding the order in which classes are presented to the class-based expansion framework. Ideally, we want to set the score of a class to a high value when it is easy to confuse.

We use a pre-trained tiny network $g(\cdot)$ to evaluate the confusion score for each class. Note the training cost of $g(\cdot)$ is much lower than that of $f(\cdot)$. To obtain the score of each class, we start by using $g(\cdot)$ to transform samples from image space into feature space and logits space:

$$g^m_x = g_f(x^m)$$

$$p^m_x = g_c(g^m_x) \quad (3)$$

where $g_f(\cdot)$ and $g_c(\cdot)$ denote the feature extractor and the classifier of the network $g(\cdot)$. Afterwards, we propose two kinds of class confusion criteria: Distance-based Criterion. To obtain the class center of each class:

$$u^m = \frac{1}{N_m} \sum_{(x, y) \in C_m} g^m_x, \quad (4)$$

where $N_m$ is the number of samples in class $C_m$. Then, the confusion score of the class $C_m$ can be reformulated as:

$$S_{dist}(C_m) = \frac{1}{N_m} \sum_{(x, y) \in C_m} \sum_{1 \leq j \leq M} \frac{||g^m_x - u^m||^2}{||g^m_x - w^m||^2}$$

$$= 1 + \frac{1}{N_m} \sum_{(x, y) \in C_m} \sum_{1 \leq j \leq M, j \neq m} \frac{||g^m_x - u^m||^2}{||g^m_x - w^m||^2}, \quad (5)$$

where $|| \cdot ||^2$ denotes the squared euclidean distance.

Entropy-based Criterion. This criterion is formulated as:

$$S_{entropy}(C_m) = \frac{1}{N_m} \sum_{(x, y) \in C_m} p^m_x \log \frac{1}{p^m_x}, \quad (6)$$

where $S_{entropy}(C_m)$ denotes the confusion score of $C_m$.

We can observe that the above two confusion criteria measure the confusion score in different spaces. The confusion score obtained by the distance-based criterion is measured in feature space, which rises as the features in a certain class move away from the center of that class and approach other class centers. The confusion score obtained by the entropy-based criterion is measured in logits space, which rises when the logits of samples of a certain class move away from the one-hot vector.

Based on the obtained scores for each class, we can get an ordered dataset:

$$D_{ord} = C_{ord_1} \cup C_{ord_2} \cup \ldots \cup C_{ord_M}, \quad (7)$$

where $ord_m$ is the index of the class with the $m$-th largest confusion score. The sorting process is detailed in Fig. 2.
C. Progressive Expansion Learning

We now describe our proposed progressive expansion learning pattern for CNN models. With the ordered dataset $D_{ord}$, we split the optimization of Eq. (2) into $K$ stages (For convenience, we set $M$ to be divisible by $K$). We start with an empty training sample pool ($D_{ord}^0 = \emptyset$). At the first stage, the first $\frac{M}{K}$ classes of the ordered dataset $D_{ord}$ are added to $D_{ord}^0$, then the training sample pool is expanded to $D_{ord}^1$:

$$D_{ord}^1 = C_{ord_1} \cup C_{ord_2} \cup \ldots \cup C_{ord_M}, \quad (8)$$

The target optimization function of $D_{ord}^1$ is:

$$\theta_1^* = \arg \min_{\theta} \sum_{(x,y) \in D_{ord}^1} l(f(x;\theta), y), \quad (9)$$

where $\theta$ is randomly initialized and $\theta_1^*$ represents the optimal model parameters learned from $D_{ord}^1$. At the $k$-th stage ($1 < k \leq K$), the training sample pool $D_{ord}^{k-1}$ is expanded to $D_{ord}^k$:

$$D_{ord}^k = C_{ord_1} \cup C_{ord_2} \cup \ldots \cup C_{ord_M} \cup \ldots \cup C_{ord_k}, \quad (10)$$

where the last $\frac{M}{K}$ classes of $D_{ord}^k$ are newly added. In order to find the optimal model parameters $\theta_k^*$ for $D_{ord}^k$, we have:

$$\theta_k^* = \arg \min_{\theta} \sum_{(x,y) \in D_{ord}^k} l(f(x;\theta), y), \quad (11)$$

where the $\theta$ is initialized by the optimal model parameters $\theta_{k-1}^*$ learned from $D_{ord}^{k-1}$.

In the simplest form of class-based expansion learning, the classes of the dataset are progressively added to the training sample pool. By analogy, using such a progressive way, we will eventually solve the problem in Eq. (2) after the samples of all the classes participate in the training process.

D. Complexity Analysis

In this section, we consider the time complexity of class-based expansion learning (CEL). Let $T_{normal}$ be the time cost for a normal training process. For CEL, if we use the same number of epochs for each stage, the time cost will be $\frac{kT_{normal}}{K}$ at stage $k$ (the ratio of the dataset size of stage $k$ to the size of the entire dataset is $\frac{k}{K}$). Then the class-based expansion learning time cost $T_{CEL}$ is:

$$T_{CEL} = \sum_{k=1}^{K} \frac{kT_{normal}}{K} = \frac{(K+1)}{2}T_{normal}, \quad (12)$$

which is a linear time algorithm.

In the experiment, we observe that reducing the number of epochs in the early stages by a factor of $\lambda$ does not sacrifice accuracy. We then train the network with the full amount of epochs only at the final stage and reducing the epoch number in other stages. In this way, the time cost $T_{CEL2}$ is:

$$T_{CEL2} = (\sum_{k=1}^{K} \frac{k}{\lambda} + 1)T_{normal} = \frac{(K-1)}{2\lambda} + 1)T_{normal}. \quad (13)$$

We can reduce the time cost of CEL by controlling the value of $\lambda$. With a large $\lambda$, only a little consumption time is required at the early stage, making the training time of our strategy comparable to the training time of normal training.

III. Experiments

A. Experimental Settings

a) Dataset: We conduct our experiments on three datasets, namely CIFAR10, CIFAR100, and ImageNet100. The CIFAR10 [27] dataset is a labeled subset of the 80 million tiny images dataset [28], which consists of 60,000 RGB images of resolution 32×32 in 10 classes, with 5,000 images per class for training and 1,000 per class for testing. The CIFAR100 [27] is similar to the CIFAR10, except that it has 100 classes containing 600 images each. There are 500 training images and 100 testing images for each class of CIFAR100. The ImageNet100 is a subset of ImageNet [1] for ImageNet Large Scale Visual Recognition Challenge 2012. It contains 129,395 training images and 5,000 validation images in the first 100 classes of ImageNet.

b) Data preprocessing: On CIFAR10 and CIFAR100, we just follow the simple data augmentation in ResNet [4] for training, including random croppping for 4 pixels padded image, per-pixel mean subtraction and horizontal flip. On ImageNet100, the augmentation strategies we use are the 224×224 random cropping and the horizontal flip.

c) Implementation details: We conduct our class-based expansion learning scheme on CIFAR10, CIFAR100, and ImageNet100 by using the state-of-the-art CNN models, including ResNet-18, ResNet-32 and ResNet-110. On CIFAR10, as described in Section II-B, we use a pre-trained ResNet-20 (trained by 60 epochs) on ImageNet to determine the order of classes. Then, as described in Section II-C, we divide the learning of the ordered dataset into 5 stages. At the first four stages, we use 60 epochs to train the network and we train the network with 300 epochs at the last stage, i.e., $\lambda = 5$. At each stage, we train the network using SGD with a mini-batch size of 128, a weight decay of 0.0001 and a momentum of 0.9. The initial learning rate is set to 0.1 and is divided by 10 after $\frac{1}{2}$ and $\frac{3}{4}$ of all epochs. On CIFAR100, we also use the pre-trained ResNet-50 to determine the order of classes. Afterwards, we divide the learning of the ordered dataset into 10 stages. At the first nine stages, we utilize 60 epochs to train the network and we train the network with 200 epochs at the final stage. The other parameters of the experiments are the same as those used on CIFAR10. On ImageNet100, we use a pre-trained ResNet-18 to determine the order of classes (30 epochs). We divide the learning of the dataset into 10 stages by the original order. At the first nine stages, we use 60 epochs to train the network and we train the network with 60 epochs at the final stage. The initial learning rate is set to 2 and is divided by 5 after 20, 30, 40 and 50 epochs. The rest of the settings are the same as those on CIFAR10. We implement our scheme with the theano [29] and use an NVIDIA TITAN 1080 Ti GPU to train the network.

B. Comparisons with the State of the Art

We compare our class-based expansion learning (CEL) with other state-of-the-art methods of Normal Training, CBS [24], DIHCL [25], and Curriculum [26]. The normal training method represents a standard training method, the CEL adopts the distance-based confusion criterion, and the CEL-2 adopts
the entropy-based class confusion criterion. The results are summarized in Table II. As shown in Table II, CEL and CEL-2 outperform other state-of-the-art methods, which demonstrates their effectiveness.

We also employ ResNet-18 to conduct experiments on ImageNet100. The results are summarized in Table III. Similar conclusions to those on CIFAR10 datasets can be made. These results demonstrate the generalization of our approach.

### C. Ablation Experiments

a) Analysis of the class order: We presented the test error results for each class of CIFAR10 in Table II and the class order in Table IV. As shown in Table IV, we can observe the cat class, the bird class, as well as the dog class, are both confusing classes defined by the distance-based confusion criterion, and the error rates of these classes are the largest ones in Table II. Table IV gives the results on CIFAR10, which shows that our method outperforms the normal training method on ResNet-32 and ResNet-110. In addition, we observe that the improvement in the performance of the model was mainly due to the preferentially selected classes (i.e., cat, bird, deer, dog, and airplane).

b) Analysis of the individual components: We carry out an experiment on CIFAR10 to analyze the individual components in the CEL method. In this experiment, without the sorted class order obtained by \( g(\cdot) \), we perform class expansion learning in a random class order, which is denoted by “w/o \( g(\cdot) \)”. The results in Table V indicate that “w/o \( g(\cdot) \)” performs better than normal training due to the class-based expansion learning process. In addition, “w/ \( g(\cdot) \)” also can improve the performance of “w/o \( g(\cdot) \)”, showing the effectiveness of the sorted class order obtained by \( g(\cdot) \).

### IV. Conclusion

In this letter, we have presented a novel class-based expansion learning scheme for CNN, which learns the whole dataset by progressively training the CNN model in a bottom-up class growing manner. By using this scheme, the classification boundaries of the preferentially selected classes are frequently stimulated, resulting in a fine-grained form. Based on the characteristics of the scheme, we have also proposed a class confusion criterion that prioritizes the classes that are easily confused. Extensive experimental results demonstrate the effectiveness of our work.

---

**Table I**

| Method               | Network | Runs | Test error (%) |
|----------------------|---------|------|----------------|
| Normal Training      | ResNet-32 | 5   | 7.08           |
|                      | ResNet-110 | 5  | 6.24           |
| CEL                  | ResNet-32 | 5   | 6.16           |
|                      | ResNet-110 | 5  | 5.71           |

| Test error of each class (%) |
|-----------------------------|
| Airplane | Automobile | Bird | Cat | Deer | Dog | Frog | Horse | Ship | Truck |
| 6.00     | 4.40       | 2.40 | 9.70| 12.40| 4.70| 4.90 | 5.00  | 4.30 | 4.70  |

**Table II**

| Dataset | Normal Training | CBS [24] | DIHCL [25] | Curriculum [26] | CEL | CEL-2 |
|---------|-----------------|----------|-------------|------------------|-----|-------|
| CIFAR10 | 30.40           | 31.49    | 31.47       | 30.02            | 29.82| 29.41 |
| CIFAR100| 7.40            | 11.80    | 4.30        | 9.60             | 2.90 | 2.00  |

**Table III**

| Dataset | Network | Normal Training | CEL |
|---------|---------|-----------------|-----|
| ImageNet100 | ResNet-18 | 29.83           | 26.86 |

**Table IV**

| Ranking | 1 | 2 | 3 | 4 | 5 |
|---------|---|---|---|---|---|
| Cat     | Bird | Dog | Airplane | Deer |

| Ranking | 6 | 7 | 8 | 9 | 10 |
|---------|---|---|---|---|---|
| Frog    | Horse | Truck | Ship | Automobile |

**Table V**

| Dataset | Normal Training | w/ \( g(\cdot) \) | w/o \( g(\cdot) \) |
|---------|-----------------|------------------|------------------|
| CIFAR10 | 7.08            | 6.16             | 6.32             |
| CIFAR100| 30.40           | 29.82            | 30.16            |

**Table VI**

| Dataset | Network | Normal Training | Test error | Test error |
|---------|---------|-----------------|-------------|-------------|
| CIFAR10 | ResNet-32 | 420             | 6.50        | 420         | 6.16        |
| ImageNet100 | ResNet-18 | 330             | 27.52       | 330         | 26.86       |

---

**Fig. 4.** The final stage of CEL method and normal training for CIFAR10 based on ResNet-32.
REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Neural Information Processing Systems, 2012, pp. 1097–1105.

[2] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[4] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[5] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.

[6] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” IEEE transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1798–1828, 2013.

[7] D. Zhang, J. Yin, X. Zhu, and C. Zhang, “Network representation learning: A survey,” IEEE transactions on Big Data, vol. 6, no. 1, pp. 3–28, 2018.

[8] H. Robbins and S. Monro, “A stochastic approximation method,” The Annals of Mathematical Statistics, pp. 400–407, 1951.

[9] D. E. Rumelhart, G. E. Hinton, R. J. Williams et al., “Learning representations by back-propagating errors,” Cognitive Modeling, vol. 5, no. 3, p. 1, 1988.

[10] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in International Conference on Machine Learning. ACM, 2009, pp. 41–48.

[11] V. I. Spikovsky, H. Alshawi, and D. Jurafsky, “From baby steps to leapfrog: How less is more in unsupervised dependency parsing,” in North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 2010, pp. 751–759.

[12] S. Basu and J. Christensen, “Teaching classification boundaries to humans,” in American Association for Artificial Intelligence, 2013.

[13] T. Zhou, S. Wang, and J. A. Bilmes, “Curriculum learning by dynamic instance hardness,” in Neural Information Processing Systems, 2020.

[14] X. Wu, E. Dyer, and B. Neyshabur, “When do curricula work?” in International Conference on Learning Representations, 2021.

[15] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” Citeseer, Tech. Rep., 2009.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” in Neural Information Processing Systems, 2012, pp. 1097–1105.

[17] A. Graves, M. G. Bellemare, J. Menick, R. Munos, and K. Kavukcuoglu, “Curriculum by smoothing,” in The organization of behavior: A neuropsychological theory. Psychology Press, 2005.

[18] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio, “Theano: a cpu and gpu math expression compiler,” in Scientific Computing with Python conference, 2010.

[19] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” arXiv preprint arXiv:1409.1556, 2014.

[20] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.

[21] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” IEEE transactions on Pattern Analysis and Machine Intelligence, vol. 35, no. 8, pp. 1798–1828, 2013.

[22] D. Zhang, J. Yin, X. Zhu, and C. Zhang, “Network representation learning: A survey,” IEEE transactions on Big Data, vol. 6, no. 1, pp. 3–28, 2018.

[23] H. Robbins and S. Monro, “A stochastic approximation method,” The Annals of Mathematical Statistics, pp. 400–407, 1951.

[24] D. E. Rumelhart, G. E. Hinton, R. J. Williams et al., “Learning representations by back-propagating errors,” Cognitive Modeling, vol. 5, no. 3, p. 1, 1988.

[25] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in International Conference on Machine Learning. ACM, 2009, pp. 41–48.

[26] V. I. Spikovsky, H. Alshawi, and D. Jurafsky, “From baby steps to leapfrog: How less is more in unsupervised dependency parsing,” in North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, 2010, pp. 751–759.

[27] S. Basu and J. Christensen, “Teaching classification boundaries to humans,” in American Association for Artificial Intelligence, 2013.

[28] T. Zhou, S. Wang, and J. A. Bilmes, “Curriculum learning by dynamic instance hardness,” in Neural Information Processing Systems, 2020.