Making Large Ensemble of Convolutional Neural Networks via Bootstrap Re-sampling

Conference Paper · December 2019

DOI: 10.1109/VCIP47243.2019.8965828

Jiawei Li
Tsinghua University
16 PUBLICATIONS **27** CITATIONS

Tao Dai
Tsinghua University
54 PUBLICATIONS **221** CITATIONS

Making Large Ensemble of Convolutional Neural Networks via Bootstrap Re-sampling

Jiawei Li, Xingchun Xiang, Tao Dai
Department of Computer Science and Technology,
Tsinghua University, Beijing, China
{li-jw15, xxc17, dait14}@mails.tsinghua.edu.cn

Shu-Tao Xia
PCL Research Center of Networks and Communications,
Peng Cheng Laboratory, Shenzhen, China
xiast@sz.tsinghua.edu.cn

***Abstract*—The ensemble of Convolutional Neural Networks (CNNs) is known to be more accurate and robust than the component CNNs models. Along with the development of a fast training method, current research has managed to make an effective ensemble of several CNNs models and require no additional training cost. However, when the ensemble size of CNNs is further increased, it is hard to observe a corresponding performance enhancement. According to the generalization capability analysis of CNNs, this phenomenon can be explained by the over-saturation of model capacity and the close correlation among the component CNNs, especially when the CNNs are trained within the same dataset. To address this problem, we propose to train CNNs on re-sampled bootstrap datasets. Extensive experiments demonstrate the bootstrap re-sampling is effective for a large ensemble size (up to 80). Besides, benefiting from the usage of the bootstrap re-sampling technique, we can also have an unbiased estimate of the standard deviation of the ensemble output.**

***Index Terms*—CNNs, Bootstrap re-sampling, Large ensemble size, Unbiased estimation of standard deviation.**

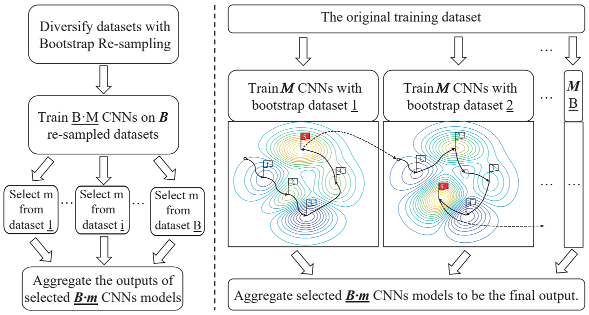
Diversify datasets with Bootstrap Re-sampling
Train B·M CNNs on ***B*** re-sampled datasets
Select m from dataset 1
Select m from dataset i
Select m from dataset B
Aggregate the outputs of selected ***B·m*** CNNs models
The original training dataset
Train ***M*** CNNs with bootstrap dataset 1
Train ***M*** CNNs with bootstrap dataset 2
M B
1
2
3
4
5
Aggregate selected ***B·m*** CNNs models to be the final output.

Fig. 1. The flow diagram (Left) and illustrated graph (Right) for learning the ensemble of CNNs via bootstrap re-sampling. In the proposed strategy, we train B×M CNNs on re-sampled datasets and then aggregate the output of selected models. As the model diversity of obtained CNNs is large, we make a large ensemble size (e.g., 80; $B = 20, m = 4$) possible.

I. Introduction

Using the ensemble of Convolutional Neural Networks (CNNs) has achieved great success on various computer vision tasks, such as crowd counting [1], facial landmark point detection [2], and object detection [3]. In the current research, the most frequent ensemble strategy on CNNs is the Multi-Column (MC) approach [4], which trains many homogeneous CNNs from scratch repeatedly and simply averages their output at the test stage. To date, despite the improvements in computer hardware enable the training of very deep single CNNs on a large dataset, training many CNNs from scratch is still a cumbersome work, which largely limits its application in practice.

The Snapshot Ensemble (SE) [5] strategy is proposed to speed up the repetitive training of many homogeneous CNNs. It uses the cyclic annealing training [6] method to quickly obtain several local optima CNNs models in only one-time training. When the ensemble size of CNNs is not large, for example, making an ensemble of m (e.g., $m \leq 8$) CNNs models, it is a very effective strategy. However, when the ensemble size is further increased, the performance enhancement will be small, or even beginning to degenerate.

Then, it is intuitive that model diversity [7] is important for further improving ensemble performance. The diversity encouraging ensemble (DEE) [8] strategy proposes to set different dropout [9] rates on different CNNs. By this way, it is able to diversify the architecture of component CNNs and enhance the diversity of obtained models to some extent. As a result, the DEE makes a larger ensemble of CNNs possible for the action recognition task. In their research, the best performance is achieved with an ensemble size of $m = 15$.

However, both of the SE and DEE strategies train all CNNs on the same one dataset. It potentially limits the possible larger value of ensemble size m. This point of view is supported by the generalization capability analysis of CNNs. On the one hand, a deep CNNs is already demonstrated to possess a capacity to memorize the entire training dataset [10]. It indicates that every component CNNs has a powerful strength. On the other hand, some studies [11]–[13] show that different component CNNs may have a close correlation. In [11], it first finds out a multi-bends low-loss pathway in the energy landscape (or, the loss surface) between two homogeneous CNNs. Then, the Fast Geometric Ensembling (FGE) [12] simplifies the form of the connecting pathway. It shows that two different CNNs can be linked in their loss surface, just with a one-bend polygonal chain or first-order Bezier curve.

This work is supported in part by the National Key Research and Development Program of China under Grant 2018YFB1800204, the National Natural Science Foundation of China under Grant 61771273, the RD Program of Shenzhen under Grant JCYJ20180508152204044, and the research fund of PCL Future Regional Network Facilities for Large-scale Experiments and Applications (PCL2018KP001).

These findings offer a clue about the failure of large ensemble size, which is highly related to the strength and correlation properties among component CNNs.

Therefore, the current ensemble strategies might have a very severe over-saturation on the model capacity, especially when the training data of every component CNNs remains unchanged. With this knowledge, to make a larger ensemble of CNNs possible, we propose to train CNNs on B re-sampled bootstrap datasets. As different dataset usually imply different solution space, this approach can achieve a larger model diversity among these component CNNs. Besides, from a statistical perspective, the bootstrap re-sampling [14] strategy can also be used for estimating the unbiased standard deviation of ensemble output. The high-level view of our framework is illustrated in Figure 1. To test the effect of the proposed method, we conduct comprehensive experiments on the CIFAR and the Tiny ImageNet benchmarks, using the ResNet and DenseNet to be the backbone networks.

It is worthwhile to highlight two key points of this paper:

- We achieve a further performance enhancement for the ensemble of CNNs, by training every CNNs models on re-sampled bootstrap datasets, even when the ensemble size is increased to a large value (e.g. 80).
- Due to the statistical usage of bootstrap re-sampling, we can estimate the unbiased standard deviation value of every ensemble output, which measures how reliable the ensemble is.

II. Preliminaries

A. Cyclic Annealing Training (CAT)

CNNs architectures such as ResNet [15] and DenseNet [16] usually have millions of parameters. A study [17] demonstrated that the more parameters, the more possible local minima could be visited in the training process. As the corresponding CNNs models of those local minima make different mistakes, the ensemble can reduce the error rates significantly. With this knowledge, the Snapshot Ensemble (SE) [5] strategy first adopts a Cycle Annealing Training (CAT) [6] method to obtain many local minima CNNs models.

Specifically, the CAT method generates many local optima CNNs models in a single training process. It abruptly raises the learning rate α and then quickly decreases it with a cosine function as below:

$$\alpha(t) = \frac{\alpha_0}{2}(\cos(\frac{\pi mod(t-1, \lceil T/M \rceil)}{\lceil T/M \rceil}) + 1), \tag{1}$$

where t is the current epoch number, T is the total epoch number, α_0 is the initial learning6 rate, and the total training epochs are divided into M cycles.

B. Bootstrap Re-sampling

In order to benefit from the ensemble, the individual predictor of bagging [18] ensemble is usually required to be uncorrelated enough [14]. Random forests [7] achieve this decorrelation with bootstrap resampled datasets and random feature subspace selection. In this paper, we introduce the bootstrap re-sampling technique to improve the ensemble of CNNs, because both of the decision tree and neural networks are low-bias and high-variance function predictor [7].

In the prediction problem, assume the original training set $D = (x_i, z_i), i = 1, ..., n$ has n samples and is composed of feature $X(x_i, i = 1, ..., n)$ and label $Z(z_i, i = 1, ..., n)$. A predictor (e.g. CNNs) $f_D(x_i) = \hat{z_i}$ can be trained on this dataset. Then, if the prediction error is within a MSE criterion (or the Cross Entropy in classification problem), it can be denoted as $err = \frac{1}{n}\sum_{i=1}^{n}(z_i - \hat{z_i})^2$. If we use the bootstrap re-sampling technique to generate B datasets $D^b = (X^b, Z^b), b = 1, ..., B$, the prediction error will be:

$$\widehat{err}^b = \frac{1}{n}\sum_{i=1}^{n}(z_i^b - \hat{z_i}^b)^2, b = 1, ..., B, \tag{2}$$

where $\hat{z_i}^b = f_{D^b}(x_i)$ is the corresponding predicted output, in which the component CNNs is trained with dataset D^b. Then, bootstrap re-sampling also brings in an unbiased estimation of the standard deviation of output:

$$\widehat{ve} = [\sum_{b=1}^{B}(\widehat{err}^b - \widehat{err}^{AVG})^2/(B-1)]^{1/2}, \tag{3}$$

where $\widehat{err}^{AVG} = \sum_{b=1}^{B}\widehat{err}^b/B$ is a simple average among these outputs. Although the accuracy is usually the primary performance criterion of a predictor, the standard error is also important for evaluating the reliability of a predictor. From the experience of statistics [14]: in complicated situations (such as ensemble of CNNs), $B \geq 25$ is usually sufficient for calculating a useful value of $\widehat{ve}$.

III. The Proposed Ensemble Strategy

A. CAT CNNs on Re-sampled Bootstrap Datasets

Our proposed ensemble strategy aims to train the component CNNs with many re-sampled bootstrap datasets fast. To achieve this, we use the CAT and bootstrap sampling techniques. This procedure is described in Algorithm 1.

Algorithm 1 The proposed ensemble strategy

```
1: Given the training set, we make $B$ bootstrap datasets, by randomly re-sampling the samples with replacement.
2: Successively train the CNNs on these datasets with the efficient cyclic annealing training (CAT) method. In every dataset, we will obtain $M$ different local optimal CNNs.
3: At the test time, select $m(m < M)$ CNNs from every dataset. Then aggregate the outputs of $B \times m$ CNNs.
4: The unbiased standard error of the ensemble output can be estimated, with the formula (3).
```

Our strategy aims to train a large ensemble of CNNs and has a corresponding performance enhancement. To achieve this, we need to trade-off the volume and diversity of the training dataset, as both of them are important for the generalization ability of CNNs. In practice, we can use a parameter t to control the size of the resampled dataset. When sample size

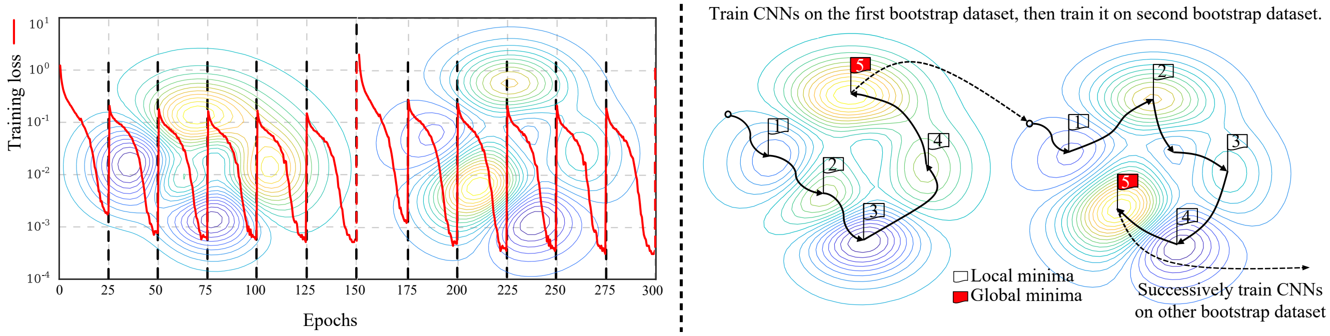

Fig. 2. A high-level illustration of the proposed strategy. **Left:** Train CNNs on two successive re-sampled bootstrap datasets with cyclic annealing parameters $T = 150$ and $M = 6$. **Right:** Then several (here we set $m = 5$) local minima CNNs will be selected to form an ensemble model. Notice that different bootstrap dataset implies different solution space.

of original dataset is denoted as n, for any sample i belonging to original dataset, the probability of sample i which does not belong to one re-sampled dataset can be represented as: $P_i = (1 - \frac{1}{n})^{(t \cdot n)}$. In the experiments, we set $t = 2.3$, then P_i is around 0.1.

B. Select Some Component CNNs for Ensemble

Once we finish the training of Algorithm 1, we need to select m component CNNs from every dataset. Theoretically, there exist $\binom{M}{1} + \binom{M}{2} + \cdots + \binom{M}{M}$ kinds of selection strategies, but several are preferred, according to the underlying model strength difference of different cycles. Addressed concretely, if we set the total training epochs $T = 120$ and $M = 6$ cycles, there are four preferred selecting methods on **every** datasets:

- selecting $m = 1$ model from cycle: 6;
- selecting $m = 2$ models from cycles: 5, 6;
- selecting $m = 4$ models from cycles: 3, 4, 5, 6; and
- selecting $m = 6$ models from cycles: 1, 2, 3, 4, 5, 6;

If we have $B = 20$ bootstrap datasets, using these methods, we will aggregate $B \times m$ = 20, 40, 80, and 120 outputs of CNNs, respectively.

IV. Experiments

A. Datasets and Setup

CIFAR. The CIFAR-10 and CIFAR-100 datasets [19] have 10 and 100 classes colored natural images, respectively. For each CIFAR dataset, there are 50,000 images for training and 10,000 images for testing, sized at 32x32 pixels.

Tiny ImageNet. The Tiny ImageNet [1] has 200 classes, in which each class has 500 training and 50 validation images. It is a subset of the ImageNet [20] dataset.

We compare the performance of our ensemble strategy with other state-of-the-art strategies, such as MC and SE. The ResNet-50 [15], DenseNet-40 [5], and DenseNet-100 [5] architectures are used to be the CNN backbones.

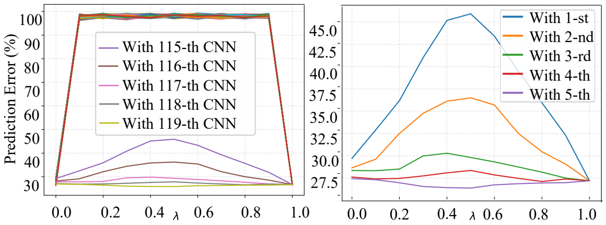

Fig. 3. **Left:** Train CNNs on 20 re-sampled CIFAR-100 datasets to obtain 120 component CNNs. The high prediction error shows the final 120-th CNNs has a low correlation to other 114 CNNs models, which are trained on different datasets. **Right:** The SE approach train CNNs on the original dataset and obtain 6 models. The smooth and low prediction error shows the 6-th model and other 5 models are close.

B. Correlation Analysis Among Component CNNs

To characterize the correlation of component CNNs, we linearly interpolate two CNNs models in their parameter space and calculate the prediction error of the interim models [21]. Specifically, assume $J(W)$ is the prediction error of a CNNs with parameters W. Then we can calculate the prediction error for a linear combination $(\lambda \cdot W_1 + (1 - \lambda) \cdot W_2)$ of two models W_1 and W_2, where λ ranges from 0 to 1, with a step value 0.1. If these two CNNs are close in the energy landscape [11] (or, high correlation), this linear combination of interim models will have low and smooth prediction error [5].

To evaluate the correlation among the component CNNs, we obtain 120 DenseNet-40 models from $B = 20$ re-sampled CIFAR-100 datasets, with the parameters $T = 120$, $M = 6$ and $m = 6$. The left subfigure of Figrue 3 illustrates the prediction error of their interim models. We can find the prediction error between the final 120-th and other 114 CNN models $f_{D^{b,cycle}}, b = 1, ..., 19; cycle = 1, ..., 6.$ is very high for any λ values. Therefore, training with different bootstrap dataset actually lowers the correlation of component CNNs.

C. Bootstrap Re-sampling Makes Large Ensemble Size

The experimental results of different ensemble strategies are summarized in Table 1. We adopt different selecting methods, as stated in section 3.2. Actually, we can compare

[1] https://tiny-imagenet.herokuapp.com

TABLE I

The classification error rate (%) of different ensemble strategies, such as MC, SE, and Our strategy (Ours).**Bold** identifies the maximum effective ensemble size of SE and Ours. **Red** means the performance beginning to degenerate. Note that only our strategy can make the unbiased standard error estimation when MC or SE can not guarantee the unbiasedness.

Traing Budget	Ensemble Size	Ensemble Strategy	CIFAR-10 ResNet-50	CIFAR-10 DenseNet-40	CFAR-100 ResNet-50	CFAR-100 DenseNet-40	Tiny ImageNet DenseNet-100
120 epochs	1×CNNs	No	7.01	5.46	28.95	24.83	39.22
	6×CNNs	SE	5.72	4.93	24.68	22.23	36.63
720 epochs	6×CNNs	MC	4.66	3.78	22.45	19.73	34.09
		Ours($m = 1$)	4.52	3.71	22.23	18.94	32.61
2400 epochs	20×CNNs	MC	3.84	3.41	20.28	18.06	33.23
	40×CNNs	**SE**($m = 2$)	3.91	3.53	21.10	19.12	33.40
		Ours($m = 2$)	3.87±0.19	3.41±0.21	20.81±0.32	18.10±0.34	32.69±0.27
	80×CNNs	**SE**($m = 4$)	4.24	3.66	21.72	19.28	33.65
		Ours($m = 4$)	**3.52**±0.17	**3.16**±0.23	**19.69**±0.28	**17.89**±0.28	**31.81**±0.26
	120×CNNs	**Ours**($m = 6$)	3.94±0.17	3.55±0.16	20.72±0.27	17.95±0.29	32.13±0.28

different strategies within the same training budget, although the component numbers might be different. It is fair because in the context of an ensemble of CNNs, more components do not always mean better performance.

Considering the maximum effective ensemble size, the performance degradation of our strategy starts from a larger ensemble size. Specifically, the SE method starts to be over-saturated when the ensemble size is increased from 40 to 80. Compared with this, our strategy is still effective with the ensemble size of 80, and it does not degenerate until further increasing to 120. This phenomenon shows that our strategy makes a larger ensemble size effective. Besides, in the table, there is an unbiased standard error estimation of our strategy. It measures how reliable the ensemble output is.

V. Conclusion

In this paper, we proposed a novel strategy to make the large ensemble of CNNs, using bootstrap re-sampling. On the one hand, the bootstrap re-sampling brings in an unbiased estimation of the standard deviation. On the other hand, the proposed strategy can address the over-saturation problem to some extent. From our experiments, the maximum effective ensemble size of component CNNs can be increased up to 80.

References

[1] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma, "Single-image crowd counting via multi-column convolutional neural network," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. IEEE, 2016, pp. 589–597.

[2] Y. Xu and S. Gao, "Bi-level multi-column convolutional neural networks for facial landmark point detection," in *European Conference on Computer Vision*. Springer, 2016, pp. 536–551.

[3] J. Guo and S. Gould, "Deep cnn ensemble with data augmentation for object detection," *arXiv preprint arXiv:1506.07224*, 2015.

[4] D. Ciregan, U. Meier, and J. Schmidhuber, "Multi-column deep neural networks for image classification," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3642–3649.

[5] G. Huang, Y. Li, G. Pleiss, Z. Liu, J. E. Hopcroft, and K. Q. Weinberger, "Snapshot ensembles: Train 1, get m for free," in *5th International Conference on Learning Representations*, 2017.

[6] J. Li, T. Dai, Q. Tang, Y. Xing, and S.-T. Xia, "Cyclic annealing training convolutional neural networks for image classification with noisy labels," in *2018 25th IEEE International Conference on Image Processing*. IEEE, 2018, pp. 21–25.

[7] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. CRC press, 2012.

[8] H. Yang, C. Yuan, J. Xing, and W. Hu, "Diversity encouraging ensemble of convolutional networks for high performance action recognition," in *International Conference on Image Processing*. IEEE, 2017.

[9] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[10] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," in *5th International Conference on Learning Representations*, 2017.

[11] F. Draxler, K. Veschgini, M. Salmhofer, and F. A. Hamprecht, "Essentially no barriers in neural network energy landscape," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, 2018, pp. 1309–1318.

[12] P. D. e. a. Garipov T., Izmailov P, "Loss surfaces, mode connectivity, and fast ensembling of dnns," in *Advances in Neural Information Processing Systems*, 2018.

[13] G. T. e. a. Izmailov P., Podoprikhin D, "Averagingweights leads to wider optima and better generalization," *arXiv preprint arXiv:1803.05407*, 2018.

[14] B. Efron and T. Hastie, *Computer Age Statistical Inference*. Cambridge University Press, 2016.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. IEEE, 2016, pp. 770–778.

[16] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. IEEE, 2017.

[17] K. Kawaguchi, "Deep learning without poor local minima," in *Advances in Neural Information Processing Systems*, 2016, pp. 586–594.

[18] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[19] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Technical report, University of Toronto*, 2009.

[20] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 248–255.

[21] I. Goodfellow, O. Vinyals, and A. M. Saxe, "Qualitatively characterizing neural network optimization problems," *arXiv preprint arXiv:1412.6544*, 2014.