Federated Learning Based Data Distillation and Augmentation for Improving the Performance of MobileNet-V2

Brijraj Singh
Indian Insitute of Technology, Roorkee
Roorkee, Uttarakhand, India
bsingh1@cs.iitr.ac.in

Durga Toshniwal
Indian Insitute of Technology, Roorkee
Roorkee, Uttarakhand, India
durgatoshniwal@gmail.com

Abstract

Overfitting is one of the most notorious and commonly occurring problems of deep neural networks (DNN), which mostly arises because of having large gape between the model's capacity and data availability. Increasing the training data to cope up with the model's capacity is one of the popular regularization methods to solve the problem. Additional training data can be generated using data augmentation method on available data. But there exists no method to guide about which samples from the available training data should be selected for augmentation purpose. Hence, randomly selecting samples is the convention. In this work, we have proposed a federated learning-based data distillation method, which provides the most appropriate samples whose augmentation improves the performance of MobileNet-V2 better than that of randomly selected samples.

1. Introduction

DNNs are known as data-hungry networks because of demanding huge data in order to learn the patterns with decent generalization performance. If DNNs are applied with limited data, it leads to the problem of overfitting. Hence, the predicting power of DNN is found restricted in the applications which lack in heuristic data. Artificially generating the data and consequently satisfying the high capacity DNN model is one of the regularization method used for better generalization performance. There have been couple of work on artificial data generation and regularization ([1], [2], [3], [4], [5], [6], [7], [8], [9]). The artificial data generation using data distillation technique is the key concept of this work. Term data distillation is used for segregating the data samples in two categories: **(a)**. Clear data points **(b)**. Crucial data points. Data samples that reside close to the class decision boundaries are referred as crucial data points and data points that reside away from the class decision boundaries are known as clear data points. The paper proposes a novel method of data distillation by exploiting the concept of federated learning performed using tiny-models. The dataset is divided among several disjoint and the definite number of data chunks. The model trained on each data chunk is referred as tiny model. Each tiny model is supposed to have the loosely fitted hyperplane built using partial observations of dataset (for instance one-tenth of total data). Since tiny models see only the partial dataset therefore, they are expected to be weak towards crucial data points. Based on this idea all the tiny models are exposed to each sample of training data and score vector (of length 10) is calculated to identify the clean, crucial data points by setting a threshold on the score. The work focuses on crucial data points as these are the points which actually contribute in making the fine decision boundaries. Hence, if crucial data points are understood well by the model then it will help in improving the model's performance. Hence, the present work generates the augmented data from the crucial data points and include them in the actual dataset. This inclusion aspires to satisfy the model's requirement of minimum data samples in order to build the most appropriate decision boundary without any loss of generality.

The novelty of the work can be highlighted as:
(1). This paper proposes a novel data distillation method for segregating the data in two parts based on their role in building the model. **(2).** In case of training the deep neural network with the restricted amount of data, the proposed method helps in taking the decision about which points should be used for data augmentation.

2. Proposed Method:

The proposed method can be categorized in following subsections and explained in Algorithm 1.

Knowledge distribution This step uses a hyperparameter k and distributes the whole training dataset $[\vec{D}]_{Tr}$ among k number of disjoint data chunks $[\vec{D}]_{Tr\_1}, [\vec{D}]_{Tr\_2} ... [\vec{D}]_{Tr\_k}$. Then a model is trained corresponding to each data chunk

Algorithm 1 Distillation_based_Augmentation($[\vec{\mathcal{D}}]$, $M\_NetV2 : \mathcal{M}, K$: *Datachunks_Count*, θ : *Threshold*)

$[\vec{\mathcal{D}}_{TR}], [\vec{\mathcal{D}}_{TE}] = [\vec{\mathcal{D}}]$
$l = length([\vec{\mathcal{D}}_{TR}])$
for $i = 0$ to k **do**
  $data\_set[i] = (\vec{\mathcal{D}}_{TR}[l/k * i : l/k * (i+1)])$
  $M_i = Train(\mathcal{M}, data\_set[i])$
  $results_i = M_i(\vec{\mathcal{D}}_{TR}.data)$
end for
$\vec{Res} = results_0||results_1||...||results_{k-1}$
$\vec{scores} = Finding\_Scores(\vec{\mathcal{D}}_{TR}.labels, \vec{Res})$
$Pruned\_data = Pruning(\vec{scores}, \theta, \vec{\mathcal{D}}_{TR}.data, \vec{\mathcal{D}}_{TR}.labels)$
$P\_A\_Data = Augmentation(Pruned\_data, Angle, Shift)$
$New\_Train\_Data = [\vec{\mathcal{D}}_{TR}] \;||\; P\_A\_Data$
$M_{new} = Train(\mathcal{M}, [\vec{\mathcal{D}}_{TR}]||[New\_\vec{Train}\_Data])$
$M_{orig} = Train(\mathcal{M}, [\vec{\mathcal{D}}_{TR}])$
$A_{orig} = \mathcal{M}_{orig}([\vec{\mathcal{D}}_{TE}])$
$A_{new} = \mathcal{M}_{new}([\vec{\mathcal{D}}_{TE}])$
return (A_{new}, A_{orig})

such that $\vec{M} = M_1, M_2, ...M_k$ in a way that each model sees only the samples belonging to the same datachunk.

Pruning This process takes into account all the tiny models of last phase and they are all exposed with entire training data. Each training sample is given to all the trained models for prediction. This gives a matrix with rows corresponding to samples of training data, where each row is a vector of size k and k is the number of prediction results. The matrix of the predicted value is than compared against the true label and the corresponding score is calculated. The score achieved by each sample uses threshold (θ) to decide whether the data sample is a clear sample or crucial sample.

Augmentation All the data samples which are categorized as crucial samples are the points which reside close to the decision boundary. In order to provide the general understanding about these points to the model, the points are transformed with rotation, shifting hence, new samples are generated.

Assimilation The data samples generated in the previous step by augmentation are appended with the training data. The method aspires to provide more general training by understanding each crucial data point and adjusting the decision boundaries accordingly.

Dataset	Original Acc (in %)	Augmentation performance		
		Data Size (in %)	Acc (in %)	
			Random Method	Proposed Method
CIFAR10	90.49	158.75	91.86	92.21
CIFAR100	65.42	194.2	68.15	69.26
CALTECH101	62.67	177.75	65.7	67.41
CALTECH256	35.8	194.7	40.8	41.77

Table 1. Augmented results in random and proposed method.

3. Results and Conclusion

The proposed work uses a novel method for data distillation and showcases its performance on four benchmark datasets (Table: 1). The method focuses on the qualitative aspect of data points, hence only specific data points (crucial) are selected for augmentation purpose. To judge the quality of each data sample, a distillation based pruning method is used for filtering out the points, which contribute less effectively in the formation of the decision boundary. The results show improvements after adding the distilled, augmented samples in the dataset. This work utilizes two augmenting techniques, **1.** Shifting $sh : 0.2$ **2.** Rotation $\alpha : 45°$. The values of these two parameters are randomly taken as $sh \in (0, 0.2), \theta \in (0, 45)°$. The proposed method generates one augment image against each selected crucial image. In this work, total number of data chunks and corresponding trainable models are fixed as 10 and pruning threshold (θ) is considered as 9.

Resources used Intel Xeon CPU, 11 GB frame buffer GPU.

Acknowledgement Thanks to Vishal Keshav, Sharan Kumar Allur and SNAP team from SRIB, India.

References

[1] Y. Gal and Z. Ghahramani. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027, 2016.

[2] 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*, pages 770–778, 2016.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[4] M. Kulkarni, K. Patil, and S. Karande. Knowledge distillation using unlabeled mismatched images. *arXiv preprint arXiv:1703.07131*, 2017.

[5] I. Radosavovic, P. Dollár, R. Girshick, G. Gkioxari, and K. He. Data distillation: Towards omni-supervised learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4119–4128, 2018.

[6] H. Scudder. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371, 1965.

[7] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1145–1153, 2017.

[8] X. Wang, A. Shrivastava, and A. Gupta. A-fast-rcnn: Hard positive generation via adversary for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2606–2615, 2017.

[9] L. Xie, J. Wang, Z. Wei, M. Wang, and Q. Tian. Disturblabel: Regularizing cnn on the loss layer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4753–4762, 2016.