proceedings

MDPI

Proceedings

A Novel Layer Sharing-Based Incremental Learning via Bayesian Optimization

Bomi Kim, Taehyeon Kim and Yoonsik Choe

Department of Electrical and Electronic Engineering, Yonsei University, Seoul 03722, Korea; bbboming@yonsei.ac.kr (B.K.); pyomu@yonsei.ac.kr (T.K.)

* Correspondence: yschoe@yonsei.ac.kr

Published: 10 November 2020

Abstract: Incremental learning means the methodology that continuously uses sequential input data to extend the existing network's knowledge. The layer sharing algorithm is one of the representative methods which leverages general knowledge by sharing some initial layers of the existing network. In this algorithm, estimating how much initial convolutional layers of the existing network can be shared as the fixed feature extractors for incremental learning should be solved. However, the existing algorithm selects the sharing configurations through not a proper optimization strategy but a brute force manner. Accordingly, it has to search for all possible sharing layer cases, leading to high computational complexity. To solve this problem, we firstly define this problem as a discrete combinatorial optimization problem. However, this problem is a non-convex and non-differential optimization problem which can not be solved using the gradient descent algorithm or other convex optimization methods, even though these methods are the powerful optimization techniques. Thus, we propose a novel efficient incremental learning algorithm based on Bayesian optimization, which guarantees the global convergence in a non-convex and non-differential optimization problem. And the proposed algorithm can adaptively find the optimal number of sharing layers via adjusting the threshold accuracy parameter in the proposed loss function. The proposed method produces the global optimal sharing layer number in only 6 iterations without searching for all possible layer cases in experimental results. Hence, the proposed method can find the global optimal sharing layer and achieve both high combined accuracy and low computational complexity.

Keywords: bayesian optimization; incremental learning; layer sharing algorithm

1. Introduction

Recently, computer vision technologies, including image recognition and object detection, have developed rapidly in the field of deep learning. Despite these remarkable achievements, one of the significant challenges in neural network-based computer vision algorithms is learning new tasks incrementally, like the cognitive process of human learning [1,2]. Three conditions are needed for the successful incremental learning algorithm:

i The subsequent data from new tasks should be trainable and be accommodated incrementally without forgetting any knowledge in old tasks, i.e., it should not suffer from catastrophic forgetting.
ii The overhead of incremental training should be minimal.
iii The previously seen data of old task should not be accessible when it is training incrementally.

For incremental learning, most previous works have focused on using knowledge from previous tasks and transferring them to a new task [3]. Sarwar et al. proposed an efficient training methodology,

called 'clone-and-branch', leveraging general knowledge from previous tasks to learn subsequent new tasks by sharing the initial convolutional layers of base networks as fixed extractors and fine-tuning in the new branch [4]. Optimal sharing layer selection is a non-convex and non-differential problem, so it can not be solved using the gradient descent algorithm or convex optimization method. Thus in [4], they explore all possible sharing layer cases to find the optimal sharing layers that meet quality specifications, and then utilize similarity score. Hence, [4] requires high computational complexity and time consumption to train all possible cases. To solve this limitation, the proposed method utilizes a Bayesian optimization (BayesOpt) to get the optimal number of sharing layer without considering all possible cases. The BayesOpt guarantees the global convergence in discrete combinatorial optimization problem [5,6]. Therefore, the proposed algorithm can find a global optimal sharing layer for layer sharing-based incremental learning.

In summary, the contributions of the proposed method are as follows:

- We firstly define the sharing layer ratio estimation problem for incremental learning as discrete combinatorial optimization problem with the global optimization strategy.
- By utilizing BayesOpt, the proposed method effectively computes the number of global optimal sharing layer without computing all possible cases.
- The proposed algorithm can adaptively find the optimal sharing layer ratio with target accuracy via adjusting the threshold accuracy parameter in the proposed loss function.
- To employ BayesOpt, the proposed objective function, which is a discrete function due to the number of layers, is designed to represent the combinatorial optimization problem with a step function as a continuous function.

2. Preliminaries

A deep convolutional neural network consists of multiple convolutional layers to extract hierarchical visual features [7]. By tracking the feature projection to these convolutional layers, the earlier layers extract the most basic part of an image, while the later layers extract much more detailed and sophisticated structures [7,8]. Incremental learning based on layer sharing technique leverages general knowledge from previously learned tasks to learn subsequent new tasks by sharing initial convolutional layers of base networks especially in a similar domain of input used in new task [3]. For efficient incremental learning, the training methodology called the *'clone-and-branch'* technique use two training methodologies [4].

First of all, to select sharing layer number, they generate an 'accuracy vs sharing' trade-off curve. Because the problem of estimating the optimal sharing layer number is a non-convex and non-differential problem, there is a large overhead for training all possible cases and determining the optimal sharing configurations that meet quality specification from this curve. Next, to get the sharing capacity of the base network for new-task, a similarity score is utilized. Random samples of each class in a new task are passed through the pre-trained base network, and the number of repeating classes is regarded as a similarity score. However, utilizing the similarity score can not be robust on randomly few sampled data, because the similarity score essentially has approximation errors on accuracy degradation in incremental learning, so is not accurate or ideal.

3. Proposed Algorithm

In this section, we explain both the proposed objective function for selecting optimal sharing layer and optimization details of the proposed algorithm through Bayesian optimization.

3.1. Combined Classification Accuracy

To measure the quality of incremental learning with n initial sharing layers, the combined classification accuracy is defined by activating combined softmax of both the base network and the new branch network. The equation of the combined classification accuracy is as follows:

$$L_{Acc}(n) = \frac{1}{N} \sum_{x_i \in D_N} n(x_i, d_i), \quad n(x_i, d_i) = \begin{cases} 1 & \text{if } F_{base,new}(x_i) = d_i. \\ 0 & \text{otherwise.} \end{cases}, \tag{1}$$

where N denotes the total number of data for testing combined classification, x_i is the i^{th} data for testing, d_i is the label of x_i, and $n(x_i, d_i)$ denotes accuracy on x_i, respectively. Therefore, if $F_{base,new}(x_i) = d_i$, which means the output of the combined network, has the same value with the ground truth on x_i. i.e., d_i, it provides 1 or otherwise 0.

3.2. Target Combined Classification Accuracy

To compute the global optimal sharing layer adaptively, the proposed method simply defines target combined classification through degraded accuracy within the baseline as the required quality specification. The baseline is the combined classification accuracy without any sharing layer of the base network.

$$L_{Target} = L_{Acc}(0) - T_{Deg}, \tag{2}$$

where $L_{Acc}(0)$ is the baseline and T_{Deg} is the threshold accuracy degradation value. Then, L_{Target} is the target combined classification accuracy. The reason of utilizing $L_{Acc}(0)$ to define L_{Target} is that $L_{Acc}(0)$ is the upper-bound value of accuracy, where every layer of the network is updated for new tasks without any network sharing.

3.3. Proposed Objective Function

The objective function $L(n)$ with sharing some of the initial convolutional layers n is the linear-combination between $L_{Acc}(n)$ and L_{Target}. The n^* is the global optimal configurations for the target combined classification accuracy degradation in the incremental learning modeling, and it minimizes the objective function. Hence, the objective function is as follows:

$$n^* = \arg\min_n L(n), \quad L(n) = ||L_{Acc}(n) - L_{Target}||_1. \tag{3}$$

However, as selecting the number of global optimal sharing layers in the discrete optimization problem, the objective function has the form of a discrete function. This form can not be applied to BayesOpt, because the Bayesian optimization guarantees the global convergence in a continuous but non-derivative combinatorial optimization problem. To solve this problem, we change the proposed objective function in a continuous step function which can be solved by Bayesian Optimization.

3.4. Global Optimal Layer Selection via Bayesopt

In order to find the optimal number of sharing layers n^*, the proposed algorithm solves this combinatorial selection problem through BayesOpt. The proposed method builds a statistical model for quantifying uncertainty using GP regression:

$$P(L(n)|L(n_{1:k})), \tag{4}$$

where $n_{1:k}$ denotes k sampled sharing layer points, and n means unsampled sharing layer point. This function denotes the posterior distribution that describes potential values at a candidate sharing layer

number [9]. Therefore, to calculate the uncertainty model from conditional distribution in Equation (4), the prior distribution $L(n_{1:k})$ is as follows:

$$L(n_{1:k}) \sim Normal(\mu_0(n_{1:k}), \Sigma_0(n_{1:k}, n_{1:k})). \tag{5}$$

Based on the prior distribution in Equation (5), the conditional distribution in Equation (4) can be recasted as follows:

$$L(n)|L(n_{1:k}) \sim Normal(\mu_k(n), \sigma_k^2(n)), \tag{6}$$

where $\mu_k(n)$ is $\Sigma_0(n, n_{1:k})\Sigma_0(n, n_{1:k})^{-1}(L(n_{1:k}) - \mu_0(n_{1:k})) + \mu_0(\hat{n})$ and $\sigma_k^2(n)$ is $\Sigma_0(n, n) - \Sigma_0(n, n_{1:k})\Sigma_0(n_{1:k}, n_{1:k})^{-1}\Sigma_0(n_{1:k}, n)$. This conditional distribution is called posterior probability distribution and quantifies the uncertainty on the unsampled sharing layer point. Following the computation of posterior distribution, the proposed algorithm uses an expected improvement (EI) acquisition function to decide the next observation points [10,11]. Therefore, the next sampling sharing layer ratio point can be defined as follows:

$$n_{k+1} = \arg\min EI_k(n), \tag{7}$$

where $EI_k(n)$ defines as $\mathbb{E}_k[\min(L(n) - L(n_k^*), 0)]$, and $L(n_k^*)$ denotes the smallest observations during k iterations. Therefore, the proposed algorithm can find optimal and accurate sharing layer point by using Bayesian optimization. In addition, because the proposed loss function is shaped in continuous function, the BayesOpt in the proposed algorithm can converge to the global optimal sharing layer number, meeting incremental learning conditions.

4. Experiment Result

In this section, we show the experimental results that demonstrate the proposed algorithm can adaptively find the optimal number of sharing layers considering some threshold accuracy degradation. And also, we compare the results of the proposed algorithm with *'clone and branch'* technique.

4.1. Implementation Details

To apply our proposed algorithm, we train ResNet50 [12] with CIFAR-100 [13] dataset. To make new classes have similar features as the old classes, we divide the CIFAR-100 to several datasets, which are chosen randomly and mutually exclusive. We train a base network with some classes, and then we update the network with remaining classes by retraining a new branch network only. When starting to retrain the branch network, we use the cloned weights of the base network instead of randomly initialized weights to have a good starting point for learning a new task.

4.2. Experimental Result

We divide CIFAR-100 into two sets, which comprise the number of classes for each set being 70 and 30. Then, we train a base network with 70 classes (T0) and update the branch new network with the rest of 30 classes (T1). Figure 1a shows the result of the selected optimal number of sharing layers through Bayesian Optimization as the threshold accuracy is 2% less than the baseline accuracy. The shaded area represents the uncertainty of unsampled points calculated through GP regression, and the black line is the posterior mean value of unsampled points. The red points denote the normalized loss value of sampled points, and the red line is drawn as the EI values based on GP regression results. In Figure 1b,c, the L2 distance of the consecutive observed points and the value of the calculated best-selected sample for every iteration are represented, respectively. Observing these results, we can find out that the 39th layer is the optimal number of sharing layers with 6 iterations. The combined classification accuracy value of the corresponding sharing layer is 67.84%, while the baseline of classification accuracy is 69.74%, as shown in Table 1. Additionally, we proceed to experiment with a different threshold value such as 3% less than the baseline accuracy, as depicted in Figure 1d–f. Thus,

we can get the optimal sharing configuration number of layers to be 47, having the accuracy of 67.03% in 6 iterations.

Table 1. Experimental results on the proposed algorithm

Task	Classes	Network	with Sharing Layers		W/O Sharing Layers
			Accuracy Degradation 2%	Accuracy Degradation 3%	Baseline
T0	70 (base)	ResNet 50: 53 convolution, 53 batch normalization, 49 ReLU, 1 average pooling, 1 FC layer	-	-	81.73%
T1	30		83.33%	81.17%	84.40%
T0-T1	100		67.84% (the optimal configuration: 39)	67.03% (the optimal configuration: 47)	69.74%

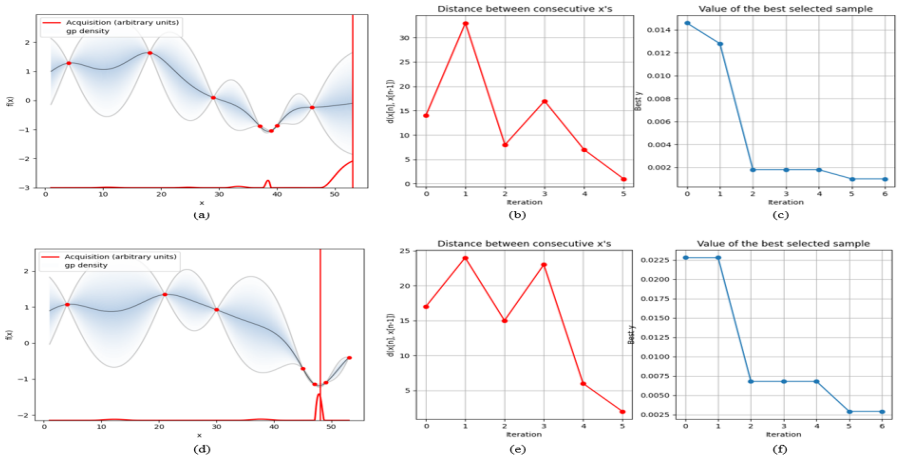

Figure 1. A visualization results of the optimal sharing layer configuration through BayesOpt. (**a**,**d**) The result of the selected optimal number of sharing layers. (**b**,**e**) The L2 distance of the consecutive observed points. (**c**,**f**) The value of the calculated best-selected sample for every iteration.

4.3. Comparison of Experimental Results for 'Clone and Branch'

We divide CIFAR-100 into three sets. The 60 classes(T0) out of the 100 classes are used for training a base network, and the rest of the 30 classes(T1) and 10 classes(T2) are used for training each incremental branch network. In Table 2, we set the threshold accuracy of 'T0-T1' to 2.33% and that of 'T0-T2' to 1.6% for making fair comparisons with *'clone and branch'* using similarity score [4]. In case of 'T0-T1', the proposed method can achieve the same accuracy result as the *'clone and branch'* in only 4 attempts. In case of 'T0-T2', we get the same result as the *'clone and branch'* in 9 attempts. As 'T2' has 10 classes, which is much smaller than 'Base', so it needs more attempts to converge because of reducing the tendency for combined classification accuracy.

Table 2. Comparison of experimental results for *'clone and branch'*.

Task	Classes	with Sharing Layers in Base Network					W/O Sharing Layers
		'Clone-and-Branch' Technique		the Proposed Method			
		Accuracy	the Optimal Layer	Accuracy	the Optimal Layer	# of Attempts	Accuracy (Basemidrule)
T0	60(base)	-	-	-	-	-	80.90%
T0-T1	60-30	66.73%	45	66.73%	45	4	68.96%
T0-T2	60-10	68.74%	46	68.74%	46	9	70.34%

5. Conclusions

In our work, we introduce a novel methodology for selecting a global optimal sharing layers for incremental learning via BayesOpt. The proposed methodology can adeptly find the number of sharing layers according to a given condition of accuracy degradation by adjusting the threshold accuracy parameter. The experimental results demonstrate that our method finds the precise sharing capacity of a base network for subsequent new tasks and converges in a few iterations. In conclusion, our proposed method is accurate and efficient. We solve the discrete combinatorial optimization problems for incremental learning by BayesOpt, which ensures global convergence.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. French, R.M. Catastrophic forgetting in connectionist networks. *Trends Cogn. Sci.* **1999**, *3*, 128–135.
2. Mermillod, M.; Bugaiska, A.; Bonin, P. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects. *Front. Psychol.* **2013**, *4*, 504.
3. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359.
4. Shakib Sarwar, S.; Ankit, A.; Roy, K. Incremental Learning in deep convolutional neural networks using partial network sharing. *IEEE Access* **2019**, *8*, 4615–4628.
5. Kim, T.; Lee, J.; Choe, Y. Bayesian Optimization-Based Global Optimal Rank Selection for Compression of Convolutional Neural Networks. *IEEE Access* **2020**, *8*, 17605–17618.
6. Kim, T.; Choe, Y. Background subtraction via exact solution of Bayesian L1-norm tensor decomposition. *IWAIT* **2020**, doi:10.1117/12.2566236.
7. Erhan, D. Visualizing higher-layer features of a deep network. *University of Montreal* **2009**, *1341*, 1.
8. Donahue, J. Decaf: A deep convolutional activation feature for generic visual recognition. *Int. Conf. Mach. Learn.* **2014**.
9. Snoek, J.; Larochelle, H.; Adams, R.P. Practical bayesian optimization of machine learning algorithms. *Adv. Neural Inf. Process. Syst.* **2012**.
10. Pelikan, M.; Goldberg, D.E.; Cantú-Paz, E. BOA: The Bayesian optimization algorithm. In Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99, Orlando, FL, USA, 17 August 1999.
11. Frazier, P.I. A tutorial on bayesian optimization. *arXiv* **2018**, arXiv:1807.02811.
12. He, K. Deep residual learning for image recognition. *Proc. IEEE Conf. Comput. Vision Pattern Recognit.* **2016**, 770–778.
13. Krizhevsky, A.; Hinton, G. Learning multiple layers of features from tiny images.*Citeseer* **2009**, *7* .