Overcoming Catastrophic Forgetting with Self-adaptive Identifiers

Fangzhou Xiong[1,2], Zhiyong Liu[1,2,3,4](✉), and Xu Yang[1,2]

[1] The State Key Lab of Management and Control for Complex Systems,
Institute of Automation, Chinese Academy of Science, Beijing 100190, China
zhiyong.liu@ia.ac.cn

[2] School of Artificial Intelligence, University of Chinese Academy of Sciences
(UCAS), Beijing 100049, China

[3] Centre for Excellence in Brain Science and Intelligence Technology,
Chinese Academy of Sciences, Shanghai 200031, China

[4] Cloud Computing Center, Chinese Academy of Sciences,
DongGuan 523808, GuangDong, China

Abstract. Catastrophic forgetting is a tough issue when the agent faces the sequential multi-task learning scenario without storing previous task information. It gradually becomes an obstacle to achieve artificial general intelligence which is generally believed to behave like a human with continuous learning capability. In this paper, we propose to utilize the variational Bayesian inference method to overcome catastrophic forgetting. By pruning the neural network according to the mean and variance of weights, parameters are vastly reduced, which mitigates the storage problem of double parameters required in variational Bayesian inference. Based on this lightweight version, autoencoders trained on different tasks are employed to self-adaptively match the corresponding task parameters to tackle sequential multi-task learning problem. We show experimentally on several fundamental datasets that the proposed method can perform substantial improvements without catastrophic forgetting over other classic methods especially in the setting where the probability distributions between tasks present more different.

Keywords: Variational Bayesian inference · Pruning · Autoencoder

1 Introduction

As a core component in artificial general intelligence (AGI), lifelong learning [1] has gradually become an essential skill to address a variety of tasks like a human being to learn. Traditional learning methods in machine learning community usually require all task data collected in advance to train the model, while it is difficult in real-world settings: tasks may not be provided simultaneously. After learning new tasks, the agent is prone to forget the old ones without accessing to previous data. This is called catastrophic forgetting in sequential multi-task

L. Cheng et al. (Eds.): ICONIP 2018, LNCS 11303, pp. 497–505, 2018.
https://doi.org/10.1007/978-3-030-04182-3_43

setting where the neural network tends to forget the weights learned in previous tasks after training on subsequent ones [2], which is a basic challenge to realize AGI.

The main dilemma that we face is to make the learned model adapt to new data without forgetting knowledge learned on the previously visited tasks. The majority of classic solutions for this problem suffer from some disadvantages. For example, fine-tune [3] behaves obliviousness property for old tasks since it only uses the optimal settings of old tasks to help initialize and study for the new tasks. As for feature extraction [4], it gives priority to reuse features obtained from the old tasks, which will present sub-optimal results for the new tasks. These methods can not achieve good performances for sequentially given tasks.

Whereas recent advances in machine learning have provided multiple ways to overcome catastrophic forgetting across a variety of domains [5,6]. Fernando et al. [7] propose an ensemble of neural networks to recombine different modules within a single network PathNet to complete different tasks. Serra et al. [8] employ a task-based hard attention mechanism to preserve previous tasks' information without affecting the current task's learning. Besides, [5] is the first to introduce Distillation Networks and fine-tune technique to enable learning without forgetting. According to this basic framework, [9] designs an extra undercomplete autoencoder to preserve the information on which the previous tasks are mainly relying. These methods, more or less, all need to specify which task to perform during the test phase, which is to say additional task identifiers have to be supplied to assign the corresponding parameters for corresponding tasks. For instance, the last fully-connected layers in [5] have to be indicated manually for corresponding tasks.

Fortunately, some other algorithms recently have received much attention, which can perform different tasks without identifiers. Elastic weight consolidation (EWC) [2], an algorithm analogous to synaptic consolidation for artificial neural networks, is proposed to reduce the plasticity of weights that are vital to previously learned tasks. It only studies a set of parameters via Fisher information to finish all tasks, which is an elegant approach in Bayesian framework to overcome catastrophic forgetting. Additionally, Lee et al. [10] present the incremental moment matching (IMM) to incrementally match the moment of Gaussian posterior distribution of different tasks in Bayesian neural networks. Nevertheless, these kind of approaches only achieve good performances on similar tasks. When the difference between task distributions becomes more larger, they are prone to forget more information about previously learned tasks.

In this paper, our motivation is to address catastrophic forgetting problem in a more realistic scenario where the gap between given tasks behaves more larger than traditional settings. For this end, a Bayesian framework is presented to remember the posterior distributions of different tasks, which actually is equivalent to an ensemble of different neural networks [11]. Meanwhile, variational inference method is introduced to approximate the posterior distributions so that each task could be trained to their optimal values. To be more specific, bottom layers near to the input are shared to catch common features between tasks, while top layers near to the output are trained individually for personalized solutions. Rather than integrating those solutions to one representation, we consider to keep their individual representations with additional task identifiers which are realized by autoencoders trained together with tasks so that they are able to help select corresponding top layers, i.e., corresponding tasks. We hence could sequentially conduct different tasks with automatically specifying parameters, which makes the algorithm behave more intelligent. More importantly, when the gap between different tasks presents more larger, the correct top layers for corresponding tasks will be selected more easily through these trained autoencoders. There is even no mismatching between autoencoders and tasks, as long as the reconstruction errors produced by each autoencoder present more different when the task distributions are great of difference. In addition, to make a lightweight network as well as select autoencoders more convenient, a pruning technique based on the mean and variance of network weights is adopted to vastly reduce the number of parameters in the network.

In summary, the main contribution of this paper is to present a self-adaptive identifier for conducting sequentially given tasks without catastrophic forgetting. Specifically, the proposed method achieves better performances when the new task is more different than previously learned tasks. The proposed algorithm can effectively utilize autoencoders with pruned weights to automatically match the corresponding parameters, enabling a more intelligent approach to perform sequential multiple tasks without indicating task identifiers so as to overcome catastrophic forgetting. The rest of the paper is organized as follows: Sect. 2 presents the proposed work. Comparative experimental results are presented and discussed in Sect. 3. Finally, concluding remarks and future work suggestions are outlined in Sect. 4.

2 The Proposed Method

The proposed method is built upon the framework of variational Bayesian inference [11], where a reparameterization trick with unbiased Monte Carlo gradients is utilized to optimize the parametric distribution. The weight w of neural network could be transformed as:

$$w = \mu + \sigma\epsilon, \tag{1}$$

where $\epsilon \sim N(0, 1)$ and $\theta = (\mu, \sigma)$ which comprises the mean μ and standard deviation σ of the network. Obviously, it requires double memory and resources than other Bayesian inference methods with neural networks, e.g. EWC algorithm, which only considers to optimize the mean of each weight.

Therefore, it is natural to decrease the overhead of the system by means of pruning the neural network based on these two parameters. More precisely, only the top layers near to the output are pruned which helps reduce the system overhead. And the bottom layers should be complete and frozen once the first task has been learned since the common features between different tasks are constructed based on these layers. At the same time, autoencoders for different

Algorithm 1. Pruned network with self-adaptive identifiers

```
Training Phase
for index $i = 1, 2, \ldots, n$ do
    Train task $i$ with stochastic gradient descent
    Train autoencoder $AE_i$, and record reconstruction error $E_i$
    if $i = 1$ then
        Frozen bottom layers forever, and record as $N1$
    end if
    Prune top layers, and record as $N2_i$
end for
————————————————————————————————————————
Testing Phase
Given a task,
    Select task identifier $j = \mathrm{argmin}_i\, E_i \ (i = 1, 2, \ldots, n)$
Perform task with network $N1 + N2_j$
```

tasks are trained with the same training data. According to the pruned network, autoencoders are utilized to help select corresponding top layers via reconstruction errors. Together with the frozen bottom layers, different networks are generated for corresponding tasks. We hence could automatically conduct different tasks without specifying which parameters to load. The proposed algorithm is presented in Algorithm 1, and we give a detailed explanation from two parts in the upcoming subsections.

2.1 Network Pruning

In this subsection, we introduce the mean and standard deviation of weights to show how to prune the neural network. Specifically speaking, the signal-to-noise ratio (SNR) is employed to address this problem, which is calculated as:

$$\mathrm{SNR} = \frac{|\mu|}{\sigma}. \tag{2}$$

By employing a large SNR which implies a large mean and a small standard deviation, we are supposed to achieve a positive effect in measuring the significance of weights so that the network will be pruned reasonably.

Given the SNR of weights w in descending order and pruning ratio k that describes the number of weights to be removed, we can initialize the mask as:

$$\begin{aligned} \lambda &= \mathrm{SNR}[length(w) \cdot (1 - \mathrm{k})] \\ \mathrm{mask} &= \mathbb{1}(\mathrm{SNR} \geq \lambda), \end{aligned} \tag{3}$$

where λ records the threshold of weights to be pruned. Then the weights can be updated with the mask:

$$w = w \cdot \mathrm{mask}. \tag{4}$$

After pruning the network, the system overhead has been decreased. If we select a high level of pruning ratio, the storage space could be vastly saved.

More importantly, the remaining weights that we care about can perform tasks with almost no performance degradation, which can be testified by subsequent experiments. Since there are multiple sets of parameters to learn for different tasks, it is more valuable to save the system overhead by pruning network.

2.2 Task Indicating

Aimed for sequential multi-task learning, bottom layers near to the input are designed to be shared between tasks. As for top layers pruned and updated by (4), we consider to match them to the corresponding tasks automatically.

Autoencoders here based on reconstruction errors are utilized for task identifiers. Furthermore, autoencoders are trained concurrently with normal task learning, and their optimal weights are usually produced by minimizing the distance between the inputs and their corresponding reconstructions. Concretely, the under-complete autoencoders (i.e., requiring the dimension of the code is smaller than the dimension of the input) are adopted to train for different tasks. In our setting, a two-layer network with a sigmoid activation function in the hidden layer is used for each task.

After training autoencoders, the minimal reconstruction error among all tasks is believed to describe the most potential autoencoder that is capable of representing the task identifier. When conducting different tasks, we could reasonably select the corresponding autoencoder as the task identifier for these pruned top layers. Actually, the automation of selecting task identifier is realized by comparing the reconstruction errors between tasks. If the gap between different task distributions presents more larger to some extent, the correct autoencoder for corresponding task will be picked out more easily, which means the proposed method could handle this situation better to guard against catastrophic forgetting.

3 Experiment

3.1 Experiment Setting

The proposed method is evaluated with a fully-connected neural network [784-800-800-10], whose first layer will be frozen after training the first task. Besides, each layer is activated by a ReLU function except the last one with softmax function used for classification, whose basic architecture is also adopted in [10]. However, we introduce the mean and standard deviation of each weight to implement the variational Bayesian learning. The datasets used in our experiments are summarized in Table 1. The MNIST dataset comprises 28×28 images of handwritten digits. The Shuffled MNIST dataset contains the same images to MNIST but whose input pixels of images are shuffled with a random permutation. The Split MNIST dataset is constructed by splitting MNIST into 0–4 and 5–9 as two tasks respectively.

Table 1. Datasets used in the experiment: name, number of classes and number of train and test samples.

Dataset	Classes	Train	Test
MNIST	10	60000	10000
Shuffled MNIST	10	60000	10000
Split MNIST (0–4)	5	30630	5105
Split MNIST (5–9)	5	29370	4895

3.2 Experimental Results and Discussion

There are two experiments conducted to show the algorithm's performance on sequential multi-task learning. One experiment employs MNIST as the first task and Shuffled MNIST as the second task, and the other invokes Split MNIST to produce two tasks respectively. The proposed method is compared with recently published algorithms EWC and IMM, which can perform tasks in a set of parameters without catastrophic forgetting.

Shuffled MNIST: As Shuffled MNIST is shuffled from MNIST, its distribution is more similar than Split MNIST with respect to MNIST. We first consider the Shuffled MNIST experiment to evaluate the proposed method. Table 2 illustrates the comparable results.

Table 2. Results of Shuffled MNIST experiment.

Method	Hyperparameter	Test accuracy
EWC	$\lambda = 20$	98.20[a]
IMM	$\alpha = 0.33$	98.30[a]
OUR	k = 0.5	98.25

[a] Optimal experimental results are cited from [10]

These approaches achieve similar results on tasks which share similar distributions. For the EWC and IMM, they address catastrophic forgetting problems in a set of parameters. For our approach, the network is pruned using SNR with the probability of 50% except the frozen layers, which results in actually only one set of parameters to learn. Instead of studying two sets of parameters for two tasks separately, this pruning technique contributes to save the storage space except two parameters (mean and standard deviation) which are inevitable overhead in variational Bayesian learning. Overall, we achieve comparable results compared to the EWC and IMM algorithms.

Split MNIST: Now we consider a more difficult situation where the distributions of two tasks present more different. First, the proposed method is evaluated

Table 3. Results of Split MNIST experiment with k = 0.5.

Method	Hyperparameter	Test accuracy
EWC	$\lambda = 20$	52.72
IMM	$\alpha = 0.33$	94.12
OUR	k = 0.5	98.58

under the same pruning ratio, and Table 3 witnesses the results with other two approaches.

In fact, the second task in Split MNIST experiment follows a more different distribution than in the Shuffled MNIST experiment. It could be easily verified in Table 3 by the test accuracy of the EWC which achieves only 52.72% compared to 98.20% in Shuffled MNIST experiment. The IMM obtains a better result since it employs mixed posteriors with transfer learning techniques. The proposed method obtains the best performance due to the individually trained autoencoders whose reconstruction errors indicate corresponding tasks to conduct. If the gap between task distributions presents more larger, the propose method will benefit more from these different distributions contrasted to other approaches. Actually, the autoencoders are employed to study reconstruction errors in different data distributions. When the task behaves more different than previous ones, its reconstruction error is more likely to be different than others. Therefore, the correct autoencoder could be picked out more easily to serve as the task identifier. Apparently, the proposed method achieves the best performance with self-adaptive autoencoders in this experiment.

Next, different pruning ratio values are designed to state the significance of SNR for pruning network. Table 4 presents the relevant results.

Table 4. Results of Split MNIST experiment with different pruning ratio k.

Pruning ratio k	0.25	0.5	0.75	0.95
Test accuracy	98.60	98.58	98.24	97.60

As the pruning ratio increases, more weights in the network are reset to zero values, which brings a more lightweight neural network. It is natural to suppose that the gradually decreased performances will present since the pruning ratio increases. However, according to Table 4, although the 95% weights of network in top layers are replaced with zero, there is still 97.60 average accuracy for the Split MNIST experiment. Obviously, the drawback in this pruned network is that more system overhead is incurred by the standard deviation of each weight in the networks. Nevertheless, it illustrates that the variational Bayesian method with pruning technique is capable of handing different tasks with almost no performance degradation. In addition, the concurrently trained autoencoders assure that the correct matching of different tasks' parameters, which is a significant step for performing sequential multiple tasks without catastrophic forgetting.

4 Conclusions

In this paper, a Bayesian framework is adopted together with variational method to approximate the posterior distributions of different tasks. In order to resolve the catastrophic forgetting problem, we utilize autoencoders as task identifiers to self-adaptively select the corresponding parameters in sequential multi-task scenario. Meanwhile, pruning technique based on SNR values contributes to a more lightweight network, which greatly benefits for the parameters reducing in traditional variational Bayesian learning. The proposed method is testified by two classical experiments, and more different tasks will be expanded to study the issue of overcoming catastrophic forgetting using the task similarity and difference in our future works.

Acknowledgments. This work was supported in part by the National Key Research and Development Plan of China under Grants 2017YFB1300202 and 2016YFC0300801, in part by the NSFC under Grants U1613213, 61627808, 61503383, 61210009, 91648205, 61702516, and 61473236, in part by the MOST under Grant 2015BAK35B00 and Grant 2015BAK35B01, in part by the Guangdong Science and Technology Department under Grant 2016B090910001.

References

1. Thrun, S.: Lifelong learning algorithms. In: Thrun, S., Pratt, L. (eds.) Learning to Learn, pp. 181–209. Springer, Boston (1998). https://doi.org/10.1007/978-1-4615-5529-2_8
2. Kirkpatrick, J., et al.: Overcoming catastrophic forgetting in neural networks. Proc. Natl. Acad. Sci. **114**(13), 3521–3526 (2017)
3. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
4. Donahue, J., et al.: DeCAF: a deep convolutional activation feature for generic visual recognition. In: Proceedings of the International Conference on Machine Learning, pp. 647–655 (2014)
5. Li, Z., Hoiem, D.: Learning without forgetting. IEEE Trans. Pattern Anal. Mach. Intell. (2017)
6. Xiong, F., et al.: Guided policy search for sequential multitask learning. IEEE Trans. Syst. Man Cybern. Syst. (2018)
7. Fernando, C., et al.: PathNet: Evolution channels gradient descent in super neural networks. arXiv preprint arXiv:1701.08734 (2017)
8. Serrà, J., Surís, D., Miron, M., Karatzoglou, A.: Overcoming catastrophic forgetting with hard attention to the task. In: Proceedings of the International Conference on Machine Learning (2018)
9. Ep Triki, A.R., Aljundi, R., Blaschko, M., Tuytelaars, T.: Encoder based lifelong learning. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1320–1328 (2017)

10. Lee, S.W., Kim, J.H., Jun, J., Ha, J.W., Zhang, B.T.: Overcoming catastrophic forgetting by incremental moment matching. In: Advances in Neural Information Processing Systems, pp. 4655–4665 (2017)
11. Blundell, C., Cornebise, J., Kavukcuoglu, K., Wierstra, D.: Weight uncertainty in neural networks. Proc. Int. Conf. Mach. Learn. **37**, 1613–1622 (2015)