Overcoming Catastrophic Forgetting by Soft Parameter Pruning

Jian Peng[†], Jiang Hao[†], Zhuo Li[†], Enqiang Guo[‡], Xiaohong Wan[↑], Deng Min[†], Qing Zhu[♯], Haifeng Li[†*]

[†]School of Geosciences and Info-Physics, Central South University

[‡] School of Civil Engineering and Transportation, South China University of Technology

[↑]Brain and Cognitive Science Research Institute, Beijing Normal University

[♯] Faculty of Geosciences and Environmental Engineering, Southwest Jiaotong University

[*]Corresponding author: lihaifeng@csu.edu.cn

Abstract

Catastrophic forgetting is a challenge issue in continual learning when a deep neural network forgets the knowledge acquired from the former task after learning on subsequent tasks. However, existing methods try to find the joint distribution of parameters shared with all tasks. This idea can be questionable because this joint distribution may not present when the number of tasks increase. On the other hand, It also leads to "long-term" memory issue when the network capacity is limited since adding tasks will "eat" the network capacity. In this paper, we proposed a Soft Parameters Pruning (SPP) strategy to reach the trade-off between short-term and long-term profit of a learning model by freeing those parameters less contributing to remember former task domain knowledge to learn future tasks, and preserving memories about previous tasks via those parameters effectively encoding knowledge about tasks at the same time. The SPP also measures the importance of parameters by information entropy in a label free manner. The experiments on several tasks shows SPP model achieved the best performance compared with others state-of-the-art methods. Experiment results also indicate that our method is less sensitive to hyper-parameter and better generalization. Our research suggests that a softer strategy,i.e. approximate optimize or sub-optimal solution, will benefit alleviating the dilemma of memory. The source codes are available at https://github.com/lehaifeng/Learning_by_memory.

1. Introduction

Humans can learn consecutive tasks and memorize acquired skills and knowledge throughout the lifetime, such as running, biking and reading. This ability, named continual learning, is also crucial to the development of Artificial General Intelligence. Deep neural networks(DNNs) have achieved remarkable success in various fields [4, 9, 28, 14], however, the existing models are unable to handle dynamic tasks and data flow because of catastrophic forgetting, i.e. networks would forget the knowledge learned from previous tasks when training on new datasets [22].methods to mitigate catastrophic forgetting have been proposed in some literatures. For instance, Rusu et al. [27], Fernando et al. [6] and Lomonaco et al. [19] attempted to restore task-specific structures of model, including some layers or modules, but this would suffer from the limitation of complex selection strategies of genetic algorithms and poor utilization of network capacity. Works [20, 25] based on rehearsal strategy reinforce previous memories by replaying experience.

An ideal learning system could learn consecutive tasks without increasing memory space, computation cost, as well as transfer knowledge from former tasks to current task. Methods of elastic parameters update[12, 26] could meet these demands by finding the joint distribution of a sequence of tasks. However, it can not restore long-term memory mainly because getting the accurate joint distribution is hard and unnecessary in a long sequence of tasks.

We propose a method to address this problem by getting the approximate solution space satisfying all tasks. It can be achieved through searching the approximate solution space from the approximate solution space corresponding to previous tasks.To achieve it, resistance based on the parameter importance are imposed on the update direction of parameters during learning new tasks.

An appropriate evaluation on parameters is expected to meet the following demands: 1) expression precision, catching some essential parameters; 2) the distribution of values is centralized and polarized to ensure the parameter space separable; 3) unsupervised. Inspired by the idea of parametric pruning, we propose a method to measure the importance of parameters based on the contrastive magnitude of the change of information entropy with and without parameter pruning. To overcome catastrophic forgetting, exerting a resistance force on updating direction of parameters with large information in the training process, to

achieve a balance point between the new task and the old one on the loss surface. The contributions of this paper are as follows:

(1) We propose SPP strategy. It requres smaller network-capacity for every single task, preserving the memory of previous tasks by constraining fewer parameters; visualization analysis shows that our method can retain important parameters according to tasks adaptively.

(2) We measure the importance of parameters by the variation of information entropy without the need of labels, rather than the variance of loss to the presence or absence of connection between units during a training progress;

(3) Experiment results show our method can effectively overcome catastrophic forgetting and improve overall performance with strong robustness and generalization ability in the case of limited network capability.

2. Related works

Model prune and knowledge distillation: Parameter pruning methods [15, 11] are based on the hypothesis that nonessential parameters have little effect on the models error after being erased and thus the key point is to search the optimum parameters that can minimum the interference to error . An effective way to narrow the representational overlap between tasks is to lessen coding parameters of representation in the continual learning model in limited capacity. Knowledge distillation pack the knowledge of complex network into a lightweight target network by the mode of teacher-student, and it also be used to tackle the problem of catastrophic forgetting.

PackNet [21] sequentially compress multiple tasks into single model by pruning redundancy parameters to overcome catastrophic forgetting. Dual memory network drew on this idea partially to overcome catastrophic forgetting by an external network. Inspired by the idea of model compression, our method utilizes parameter-importance to set up a soft mask rather than hard pruning based on binary mask, it dose not completely truncate the unimportant parameters, but adaptive adjust to later tasks to some extent, shares part of parameters among multiple tasks and save model capacity compared to hard pruning as well as enjoy lowered performance penalties.

Regularization strategies: Those Methods reduce representational overlap among tasks to overcome catastrophic forgetting by regularization such as weights freezing and weight consolidation.

Weight Freezing, enlightened by distributed encoding of human brain neurons, tries to avoid overlaps between crucial functional modules of tasks. For instance, Path-Net [6] sets up a huge neural network, then fixes specific function module of network to avoid being interfered by later tasks. Progressive Neural Network (PNN) [27] allocates separate networks for each task and performs multitasks by progressive expansion strategy. This kind of methods fix important parameters of a task durably to prevent network from forgetting acquired knowledge. However, those methods suffer network capacity exploding from long-term tasks.

A classic weight consolidation method [1, 31] is elastic weight consolidation (EWC) [12]. EWC, inspired by the mechanism of synaptic plasticity, updates parameters elastically via determining important parameters. This type of method encode more tasks knowledge with less network capacity and lower computation complexity compared with Path-Net and PNN. The upper bound of tasks EWC can learn is constrained by capacity of network which is determined by the model structure. Since the model structure is invariant during learning process, the increased tasks potentially lead to performance of degenerate of EWC.

3. Methodology

3.1. Motivation

The cause resulting in catastrophic forgetting is the drift of local minimum point when training a new task. We claim that it is feasible to approximate a distribution satisfying all tasks through seeking for the current solution space from that of previous tasks. It can be achieved by imposing resistance on parameters in proportional to their importance.(Figure 1). The key is to ensure the sparseness and representation precision of parameter importance, but there are still some problems in the previous methods of measuring parameter importance:

(1) Getting importance value of parameters by calculating gradient descent of loss function would unavoidably underestimates these importance value as the model reaches its convergence;

(2) Supervised, it relies heavily on labeled training data and testing data;

(3) The distribution of important parameters is relatively divergent. Calculating the importance of parameters depends on sensitive of a model responding to the parameters perturbation instead of the magnitude of parameters weights. Lower magnitude could possibly result in higher model sensitivity than higher ones , without considering the effects of cumulative changes in parameters, resulting in a high occupancy of capacity for each task.

We design a method to measure the parameter importance satisfying the concentrated and polarized distribution, and then propose a framework to overcome catastrophic forgetting by SPP.
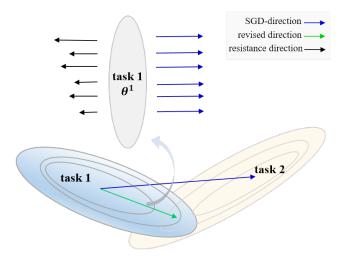
Figure 1. Weight consolidation for overcoming catastrophic forgetting. Blue arrow denotes standard SGD optimizer; black arrow is the resistance about the memories of previous tasks on parameters and green denotes a revised direction of SGD-derection, under the constraint of resistance direction.
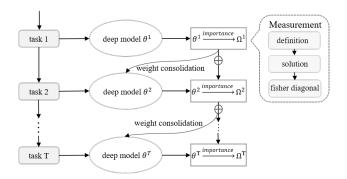

Figure 2. Framework for continual learning on T sequential tasks.

3.2. Framework

Following 3.1, we present the framework of SPP strategy in Figure 2. We calculate the the coefficient of resistance of parameters on previous $T-1$ tasks after learning the$(T-1)^{th}$ task. Then when the Tth task is coming, we update the direction of gradients according to the former coefficient of resistance.

3.2.1 Measure of importance of parameter

Definition: Given a well-trained model, we try to train parameters W on input X to reduce the error $E = \sum_{i=1}^{C} p_i \log q_i$, and the model learned can be expressed as $F(X, W) \to E$. If we set W_k as 0the change of error δE corresponding to W_k can be written as $F(X, W, 0) - F(X, W, W_k) \to \delta E$. The larger δE means the more important W_k. The formula of Taylor expansion is:

$$\delta E = (\frac{\partial E}{\partial W})^T \delta W + \frac{1}{2} \delta W^T H \delta W + O(\|\delta W\|^3) \quad (1)$$

$H \equiv \partial^2 E / \partial W^2$ is the Hessian matrix on parameters; $\partial E / \partial W$ represents the gradient on the W. The gradient will close to 0 when the model converges and the first item on the right side will be too minimal to calculate a precise value of the error change to the parameter perturbation. Therefore second-order approximate solution is used as a satisfying amendment.

Solution - The method above depends on the true distribution p which is calculated by labeled data. To get rid of the need of labels, we use the information entropy approximate the error E because the distribution p and predicted distribution q are proximate on a well-trained model. Thus, the parameter importance is in term of:

$$\delta E = (\frac{\partial H(q)}{\partial W})^T \delta W + \frac{1}{2} \delta W^T H \delta W + O(\|\delta W\|^3) \quad (2)$$

Where $H(q) = \sum_{i=1}^{C} q_i \log q_i$. The idea behind is to measure the steady state of a learning system utilizing information entropy. We explain it that: the output distribution of model will gradually involve from a random state into a stable state and with entropy decreasing. When the model converges, the system will perform stably on the training data, which means low entropy and certain output distribution. And the statement of parameters connection would influence the stability, which means the change of entropy.

Hessian diagonalization - The calculation of Hessian is complex and high computation. We introduce the Diagonal Fisher Information Matrix [24] to approximate Hessian matrix. The advantage is that its computational complexity is linear and can be solved quickly through gradient. However, the diagonalization may lead to a loss of precision. We speculate that better results can be obtained if we adopt a better Hessian approximation method.

3.2.2 Cumulative importance computation

Given a set of t+1tasks, we calculate the importance $\Omega_{i,j}^t$ of the parameter $w_{i,j}$ after learning the tth task. Where i and j represent the connection between the ith neuron and the jth neuron in neural networks respectively.

$$\Omega_{i,j}^t = max(0, \Omega_{i,j}^t) \quad (3)$$

According to equation (2), the positive values of parameter importance indicate these parameters are significant to current task and vice versa. Thus, We set the negative value of importance all to 0 to reduce the resistance of learning new tasks. After learning the t+1 task, we accumulate the

Table 1. Pseudo-code for overcoming catastrophic by soft parameter pruning

---

Overcome Catastrophic Forgetting by Soft Parameter Pruning:

Start with:
  $W_{i,j}^*$: old task parameters
  $W_{i,j}$: new task parameters
  X, Y: training data and ground truth on the new task
  tasks: all tasks numbers
  $H(q)$: information entropy of output
  H: Hessian Matrix
  e_k^T: the unit vector with correspongding to W_k
Training:
  for task$\in$ tasks do
    $W_{i,j}^*$.assign($W_{i,j}$)   //Update old task parameters
    //Calculate the importance of the parameters of the T-1 tasks
    $\Omega_{i,j}^t = max(0, (\frac{\partial H(q)}{\partial W_{i,j}})^T \delta W_{i,j} + \frac{1}{2} \delta W_{i,j}^T H \delta W_{i,j})$
    $s.t. e_k^T \delta W_{i,j} + W_k = 0$
    $\Omega_{i,j}^{1:t} = \Omega_{i,j}^{1:t-1} + \Omega_{i,j}^t$ //Cumulative importance computation
    Define: $\hat{Y} = CNN(X, W_{i,j})$ //new task output
    $W_{i,j} \leftarrow arg_{W_{i,j}} min(L_{new}(Y, \hat{Y}) + \lambda \sum_{i,j}^W \Omega_{i,j}^{1:t}(W_{i,j} - W_{i,j}^*)^2)$
      //Update new task parameters
  end for

---

importance of the previous tasks to obtain the accumulated importance on the parameters on t+1 tasks:

$$\Omega_{i,j}^{1:t+1} = \Omega_{i,j}^{1:t} + \Omega_{i,j}^{t+1} \qquad (4)$$

3.2.3 Weight consolidation

To avoid previous memories forgetting, we protect important parameters from being destroyed in subsequent training process by adding additional regular terms in the target function:

$$L = L_{new} + \lambda \sum_{i,j}^W \Omega_{i,j}^{1:t}(w_{i,j} - w_{i,j}')^2 \qquad (5)$$

L_{new} is the loss function of the current task. Here, $w_{i,j}'$ is the parameter of the model after learning last task, $w_{i,j}$ denotes the parameter corresponding to current task, and $\Omega_{i,j}^{1:t+1}$ denotes the cumulative importance of a parameter in the previous t tasks.And we present roughly our algorithm in Table 1.

4. Experiment and analysis

4.1. Experiments setting

Data. The permuted MNIST [29] or Split MNIST [16] is too simple to evaluate our methods. In order to verify the ability of generalization, we tested the proposed method on three tasks: image classification task with CNN model, long-term incremental learning and generative task with Variational AutoEncoder (VAE) model.

In image classification task, the Cifar10 [13], the NOT-MNIST [2], the SVHN [23], and the STL-10 [3] which are

all RGB images with the same size of 32*32 pixels, are chosen. In long-term incremental learning task, Cifar100 [13] is used for medium scale network model, and Caltech101 [5] is used for large scale network model (shown in supplement). In the generative task, celebA [18] and anime face crawled from web are selected as test data. These two database share the same resolution.

Baseline We compared our method with state of the art methods, including LWF [17], EWC [12], SI [31] and MAS [1], as well as some classic methods, including standard SGD with single output layer (single-head SGD), SGD with multi-output layers, SGD with the intermediate layers frozen(SGD-F), and fine-tuning intermediate layer(finetuning). We defined a multi-tasks joint training with SGD (Joint) [30] as the baseline to evaluate the difficulty of a sequential tasks.

Evaluation We utilize Average Accuracy(ACC), Forward Transfer (FWT), and Backward Transfer (BWT) [20] to estimate the model performance: (1) ACC, evaluating the average performance of processing tasks; (2) FWT, describing the suppress of former tasks on later tasks; (3) BWT, describing the forgetting of previous tasks. Evaluating the difficulty of individual task through testing the model by multi-tasks joint training [30] is more objective than testing the model of single task. Therefore we put forward a modified version. Given T tasks, we evaluate previous t tasks after trained on the tth task. Denoting the result of i task tested on the jth task model as $P_{j,i}$. We use three indicators:

$$ACC(i) = \frac{1}{T} \sum_{i=1}^T P_{T,i} \qquad (6)$$

$$FWT = \frac{1}{T-1} \sum_{i=1}^{T-1} P_{i,i} - m_i \qquad (7)$$

$$BWT = \frac{1}{T-1} \sum_{i=1}^{T-1} P_{T,i} - P_{i,i} \qquad (8)$$

Higher value of ACC indicates better overall performance, and higher value of BWT and FWT indicate better trade-off between memorizing previous tasks and learning new ones.

Training All models share the same network structure with dropout layer[8], and we initialized all parameters on MLP with random Gaussian distribution which has the same mean and variance ($\mu = 0, \sigma = 0.1$), and applied Xavier on CNN. We optimized models by SGD with initial learning rate searching from $0.1, 0.010.001$ with a decay ratio of 0.96, and with uniform batch size. We trained models with fixed epoch and global hyper-parameters for all tasks. We chose the optimal hyper-parameters by greedy search.

Table 2. The results of Split-MNIST

Method	FWT(%)	BWT(%)	ACC(%)
SGD	-0.31	-34.01	61.53
SGD-F	-18.6	-12.9	84.82
Fine-tuning	**-0.29**	-13.9	82.04
EWC [12]	-4.99	-6.43	88.75
SI [31]	-6.19	-3.51	90.67
MAS [1]	-4.38	-2.08	94.09
LWF [17]	-4.42	-2.04	94.08
Joint [30]	/	/	**99.87**
Ours	-0.44	**-0.75**	98.31

Table 3. The results of permuted MNIST of 10 tasks

Method	FWT(%)	BWT(%)	ACC(%)
SGD	1.11	-18.05	70.45
SGD-F	-14.90	**0.10**	81.99
Fine-tuning	0.75	-6.21	80.69
EWC [12]	-0.98	-2.57	91.97
SI [31]	-0.56	-4.40	90.21
MAS [1]	-1.23	-1.61	92.6
LWF [17]	0.67	-24.02	74.15
Joint [30]	/	/	**95.05**
Ours	**2.33**	-3.22	94.51

4.2. Experiment results and analysis

4.2.1 MLP& MNIST

Split MNIST

We divided the data into 5 sub-datasests, and trained an MLP with 784-512-256-10 units. In Table 2, we present experiment results with split MNIST. Not all continual learning strategies work well on all indexes. Fine-tuning and SGD perform best on FWT, because no free memories are required for the subsequent tasks, and some features may be reused to improve learning of the new tasks while tasks are similar. LWF, MAS and SI perform well on BWT and ACC, and our method achieve the best performance on both indexes besides the joint learning method. We conclude that the model learns the general features from multiple datasets, which means models implicitly benefit from data augmentation. Our results of ACC and FWT can rival the best ones in single index and our model has the least catastrophe forgetting problem on BWT and it has only a 1.5 reduction in ACC after learning 10 tasks. In general, our method outperforms another eight approaches.

Permuted MNIST We evaluate our method on 10 permuted MNIST tasks. In Table 3, we present the results of our approaches and those of others. As what we expected, our method performs best on FWT, which are superior to SGD, we own it to the possibility of some features of lower layer can be shared by new task and there is enough capacity to relieve the pressure on capacity demand in new tasks. SGD-F gets the highest score on BWT, because SGD-F has fixed parameters, which help protecting the parameters of previous tasks from being overwritten, but it is at cost of the disability to learn new tasks flexibly. LWF performs worst on permuted MNIST compared with the split MNIST despite a good score, which may be attributed to dataset changing mentioned above on FWT. And our method gets the comparable performance on ACC.

4.2.2 CNN & image recognition

Sequence of image recognition tasks

Further, we test out method on nature visual datasets based on VGG [28] with 9 layers and batch normalization layer to prevent gradient exploding. Specifically, we train and test on MNIST, notMNIST, SVHN, STL-10 and Cifar10 in this order, which have been processed to the same amount of train images and categories (50,000 and 10, respectively).

Overall, our method achieved the best performance on FWT, BWT and ACC. As Figure 3 shown, our method is almost one-third of LWF and MAS on FWT. It indicates that our proposed method work well on alleviating the dilemma of memory, and the test accuracy is close to the baseline nearly. Namely, our method drives the network to train well on the sequence of tasks and on BWT, ours also reaches the top result, which means that it ensures the network still have a good ability to handle the old tasks after the continual training. On ACC, our method has also achieved nearly performance as multi-task joint training, which shows that networks can do a good trade-off among tasks. The result of Fine-tuning is better than that of SGD, indicating that the configuration of independent classifier for each task can prevent catastrophic forgetting to some extent. We speculate that it is because the features of different tasks at the high layer are highly entangled, and using different classifiers can alleviate this situation a little.

4.2.3 Robust analysis

To test the stability of our method to hyper-parameter, based on the above experiment, we test the method under different λ. The result shows that ours have a strong robustness to hyper-parameter in a large range of values and can overcome strophic forgetting to some extent. As shown in Figure 4, when λ is 0.01, the network only focuses on the training on the new task and does not care about the protection of the old task. In this case, all the three indicators are extremely poor, and the proposed method and SGD are almost
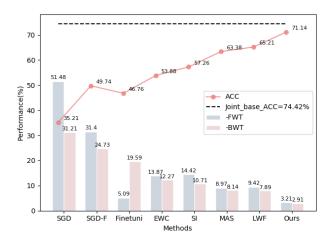
Figure 3. The performance of different overcoming catastrophic forgetting methods on the sequence of visual datasets. The method based on regularization produces a little effect starting from EWC, although the effect is limited; MAS and LWF are close; Our method reaches the best performance on all the indicators.
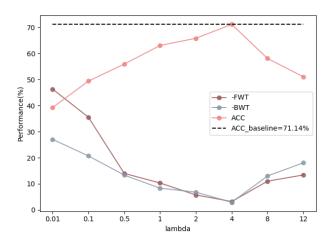

Figure 4. The performance of SPP under different hyper-parameter. The horizontal axis represents the value of hyper-parameter. The vertical axis represents the result of three indicators. The dotted black line indicates the baseline of accuracy.

the same at this time. When λ reaches 0.1, the proposed method has achieved relatively good performance and has greatly improved on all three indicators. If λ is in the range of 0.5 to 4,the performance is relatively stable. The proposed method achieves the best performance with $\lambda = 4$. As λ continues rise, the network memorizing too much results in lack of capacity to learn, which makes the performance of new task processing decreased.

4.2.4 Continual learning in VAE

To test the generalization of our method, we apply it in VAE, we carry out tasks in sequence from human face to anime face. We resize the samples of two datasets to the same size of 96*96, and train a VAE with conv-conv-fc encoder layer and fc-deconv-deconv layer on both sides. Then we use separate latent variable to train single task, which is essential for the performance of VAE because of significant difference of distributions between two tasks.

We trained models by three manners: (1)training on the Celeba dataset from scratch; (2) training on the Celeba and then training on the anime face with SGD; (3) training on the Celeba and then training on the anime face with SPP.

In Figure 5, we present the samples of human face produced by three models. The results show that our approach can well preserve the skill of human face generation while learning anime face. The model with SPP works well as the model train on the Celeba, but the model with SGD loses the ability. it p roves that SPP has strong generalization.

4.2.5 Discussions

Analysis of parameter-importance. As mentioned above, we expect the distribution of parameter-importance is concentrated and polarized. In Figure 6, We show the distribution of parameter importance obtained by the three methods. The results shows that a distribution with these two characters contributes to overcoming catastrophic forgetting. The left figure shows the our distribution is sharp at low importance and high importance, indicating that our method frees more parameters to learn more tasks. And the figure on the right shows similar results based on CNN and CIfar10. Compared with other methods, the distance between peaks of the distribution is closer shown in the middle figure. We speculate the absolute distance at is large enough(0-0.4) to distinguish different tasks. As shown in the right figure, the overall parameter importance is low in values, we believe that the capacity of vgg9 for MNIST is sufficient enough, and compared with other methods, our method is also polarized.

Parameter space similarity and changing analysis. We carried out six sequential tasks using our method with Permuted-MNIST and analyze the experiment results in comparison with SGD of single-head and Fine-tuning of multi-head as control groups:

(1) The evolving of overall average accuracy is shown in Figure 7(a) which indicates that our method is more stable and achieves better results as growing number of tasks;

(2) In order to verify model can preserve previous memory efficiently, we use Frchet distance [7] to measure

| (a) | (b) | (c) |

Figure 5. Overcome catastrophic forgetting from face dataset to anime dataset use VAE. To guarantee the objectivity of the results, we respectively utilize different data and different network structures. left: The test sample of human face with generator from the human face dataset; middle: The test sample of human face with generator after trained from celebA to anime face dataset, without using our approach; right: the test sample of human face with generator after trained from celebA to anime face dataset using our approach.
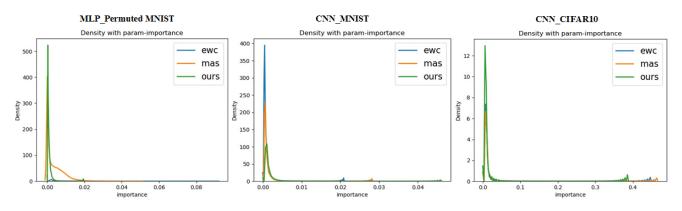

Figure 6. Distribution of parameter-importance. The horizontal axis is importance value; the vertical axis is density; the blue solid line is the calculation result of Fisher informationmatrix in EWC; the orange solid line represents the method of MAS; the green solid line is the result of our method. Left: the distribution of parameter-importance measure on MLP model trained with Permuted-MNIST; mid: the result measure on vgg9 trained with MNIST; right: the result measure on vgg9 trained with CIFAR10

the similarity of parameter importance distribution between the first tasks and the last tasks, Figure 7(b). In general, F value in our method is far greater than other two methods, which indicates that our method can better retain the important parameters of previous task, and the F values are greater in deeper layers of networks, which reveals that strengthening protection of parameters on deep layers may tremendously help tackle catastrophic forgetting;

(3) In Figure 7(c), We utilized weighted sum of squares of difference between the first and the last task to measure the parameters' change. Our results show that parameters in deeper layers change less, and the fluctuation of parameters based on our methods is much larger than

other methods. It indicates our method can preserve the former memories but it leads to larger network capacity to learn new tasks.

Visualization analysis. We visualize the negative of absolute value of parameters change(left), and compare it with the distribution of parameter importance(right). Our results show that our method can prevent significant parameters from being updated and make full use of non-significant parameters to learn new tasks. In Figure 8, in the black dotted bordered rectangle of the 1st row, parameters with warm color change little. In contrast, parameters in the second column of the picture on right are unimportant and they change hugely. It indicates that our method can precisely capture the significant parameters and prevent them from
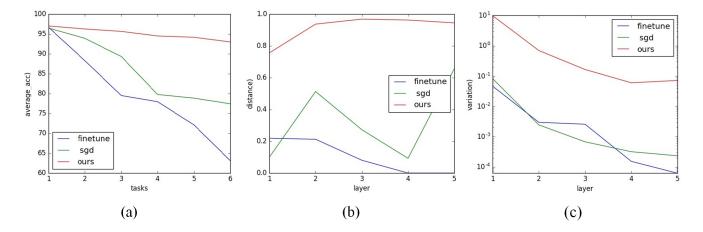
Figure 7. Parameter space similarity and changing analysis on Pemuted-MNIST sequential tasks, a red line denotes our method, a blue line denotes fine-tuning and a green line denotes standard SGD with single head; (a): Overall average accuracy in 6 permuted MNIST sequential sub tasks; (b): Similarly of parameter space; (c): Parameter variance between parameters of tasks.
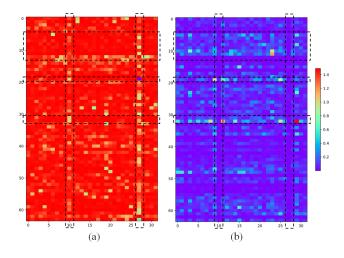

Figure 8. Visualization of importance and variance of parameters. The horizontal axis represents the neurons of the output layer, the vertical axis represents the neurons of the input layer, and each element represents the connection between the neurons of the input and output layer. Left: variance of parameters between two tasks, the colder the color is, the smaller the variance is; right: importance of parameters of the first task, the warmer the color is, the more significant the parameter is.

being updated to prevent forgetting.

5. Conclusion and future works

In this paper, we proposed a Soft Parameters Pruning (SPP) strategy to overcome catastrophic forgetting by find a trade-off between short-term and long-term profit of a learning model. Our strategy tries to free those parameters less contributing to remember former task domain knowledge to learn future tasks, and preserve memories about previous

tasks via those parameters effectively encoding knowledge about tasks at the same time. The SPP strategy also catches parameters with high information and prevent them from being overwritten in a soft way to prevent forgetting. Experiments show some advantages of SPP strategy:

(1) Defining a measurement strategy guaranteeing the precision;

(2) Our approach is low-sensitive to hyper parameters;

(3) Our approach can be extended to generative model.

Our evidences suggest that that finding a approximate optimize or sub-optimal solution will benefit alleviating the dilemma of memory. We also find that concentration and polarization properties of parameters distribution are significant for overcoming catastrophic forgetting.

The aim of overcoming forgetting in long-sequence tasks has not been fully achieved because of protecting some parameters through measurement based on single strategy is not entirely convincing. We suggest that well-structured constraints to control parameters behavior or well-designed pattern of parameters distribution may be crucial to the good performance of a model to overcome forgetting. Also, research on human brain memory is considering a potential way to solve this problem [10]. The problem of overcoming catastrophic forgetting is still open.

References

[1] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars. Memory aware synapses: Learning what (not) to forget. *arXiv preprint arXiv:1711.09601*, 2017.

[2] Y. Bulatov. Notmnist dataset. *Google (Books/OCR), Tech. Rep.[Online]. Available: http://yaroslavvb. blogspot. it/2011/09/notmnist-dataset. html*, 2011.

[3] A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223.

[4] J. Deng, W. Dong, R. Socher, and L. J. Li. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255.

[5] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE transactions on pattern analysis and machine intelligence*, 28(4):594–611, 2006.

[6] C. Fernando, D. Banarse, C. Blundell, Y. Zwols, D. Ha, A. A. Rusu, A. Pritzel, and D. Wierstra. Pathnet: Evolution channels gradient descent in super neural networks. *arXiv preprint arXiv:1701.08734*, 2017.

[7] M. M. Fréchet. Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, 22(1):1–72, 1906.

[8] I. J. Goodfellow, M. Mirza, D. Xiao, A. Courville, and Y. Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

[9] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, and A. Coates. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

[10] D. Hassabis, C. Summerfield, D. Kumaran, and B. Matthew. Neuroscience-inspired artificial intelligence. *Neuron*, 95(2):245–258, 2017.

[11] B. Hassibi and D. G. Stork. Second order derivatives for network pruning: Optimal brain surgeon. In *Advances in neural information processing systems*, pages 164–171.

[12] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, and A. Grabskabarwinska. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences of the United States of America*, 114(13):3521, 2016.

[13] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Report, Citeseer, 2009.

[14] Y. Lecun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436, 2015.

[15] Y. LeCun, J. S. Denker, and S. A. Solla. Optimal brain damage. In *Advances in neural information processing systems*, pages 598–605, 2015.

[16] S.-W. Lee, J.-H. Kim, J. Jun, J.-W. Ha, and B.-T. Zhang. Overcoming catastrophic forgetting by incremental moment matching. In *Advances in Neural Information Processing Systems*, pages 4652–4662.

[17] Z. Li and D. Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[18] Z. Liu, P. Luo, X. Wang, and X. Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15:2018, 2018.

[19] V. Lomonaco and D. Maltoni. Core50: a new dataset and benchmark for continuous object recognition. *arXiv preprint arXiv:1705.03550*, 2017.

[20] D. Lopez-Paz. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476.

[21] A. Mallya and S. Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. *arXiv preprint arXiv:1711.05769*, 1(2):3, 2017.

[22] M. McCloskey and N. J. Cohen. *Catastrophic interference in connectionist networks: The sequential learning problem*, volume 24, pages 109–165. Elsevier, 1989.

[23] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5.

[24] R. Pascanu and Y. Bengio. Revisiting natural gradient for deep networks. *arXiv preprint arXiv:1301.3584*, 2013.

[25] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. icarl: Incremental classifier and representation learning. In *Proc. CVPR*.

[26] H. Ritter, A. Botev, and D. Barber. Online structured laplace approximations for overcoming catastrophic forgetting. *arXiv preprint arXiv:1805.07810*, 2018.

[27] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

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

[29] R. K. Srivastava, J. Masci, S. Kazerounian, F. Gomez, and J. Schmidhuber. Compete to compute. In *Advances in neural information processing systems*, pages 2310–2318.

[30] X.-T. Yuan, X. Liu, and S. Yan. Visual classification with multitask joint sparse representation. *IEEE Transactions on Image Processing*, 21(10):4349–4360, 2012.

[31] F. Zenke, B. Poole, and S. Ganguli. Continual learning through synaptic intelligence. *arXiv preprint arXiv:1703.04200*, 2017.

Supplementary Materials

6. Incremental learning

Large scale dataset from Caltech-101 To test the performance of our method on a larger dataset, we randomly splitted the Caltech-101 into 4 subsetswith 30,25,25,22 classes respectively, and then divided each part of them into training and validation set according to the ratio of 7:3. In the experiment, we resized the images to [224,224,3], normalized the pixels into [0,1] and randomly flipped the images left and right to augment the data in preprocess. We employed ResNet18 as the basic network. Because the categories of four datasets are not consistent, we added a new separate classifier and a fully connected layer before the classifier for each task. Each new fc layer has 2048 neural units and the dropout rate is set to 0.5. The iteration size and batch size of every task are 100 epochs and 128. The initial learning rate is set to 0.001, using decay with every 100

epochs to 90% of the original. In order to prevent overfitting, we search the hyper parameter randomly ranging from 0.5 to 30. Due to the inconsistent numbers of categories in four subsets, we do not compare our method with SI.

A well-functioning model is expected to be stable under abrupt change of tasks. In order to evaluate the stability of model on unseen tasks, we designed an indicator SMT as follow:

$$SMT = \frac{1}{T-1} \sum_{j=1}^{T-1} D_j \qquad (9)$$

Where D is the variance of one task for sequential learning, which reflects the performance fluctuations of a task.

Long sequence for CIFAR100. As shown in Figure 9, all current methods do not perform well in large scale datasets as the number of tasks increasing. In the fourth task, the ACC of our method is less than that of SGD-F. However but outperform EWC, MAS and LWF. In SMT, when the model learns the second and third task, our method is worse than SGD and MAS. When it comes to the fourth task, our method works better than all the rest methods in BWT and SMT, which shows that our method can keep the good memory of tasks with longer sequence and was better stability. In FWT, our method shows the best performance except the MAS. Overall, our method is better than state-of-the-art methods based on regularization.

The results in Figure 10 indicate that it is still difficult to make models capable of long-term memory, especially in complex tasks. Our method achieved similar results in the overall performance among regularization methods, but SGD-F and finetuning performed better when the number of learning task is large, and LWF almost lost its learning ability. On BWT, our method and MAS achieved a better result. SGD-F performed best on preventing forgotten, because the weights were completely fixed. Noting that LWF shows higher BWT, but the data is useless due to the loss of learning ability. On FWT, our method achieved the best results, which indicating that our method has little impact on the learning of new tasks while preserving previous knowledge.
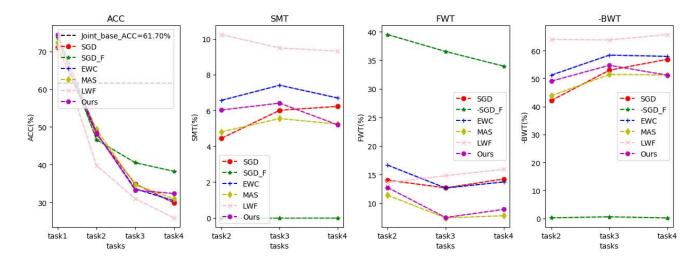
Figure 9. Performance for subset of Caltech101. The x-axis denotes the tasks trained on resnet-18 and y-axis denotes the indicator of ACC, SMT, FWT and BWT, noting that we presented the negative value of the FWT and BWT in figures.
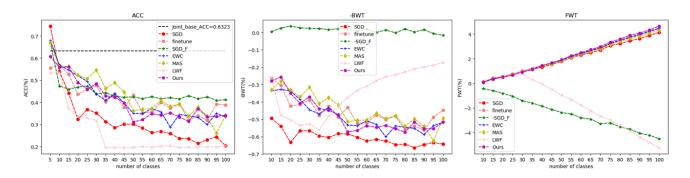

Figure 10. Performance for incremental learning on CIFAR100. The x-axis denotes the tasks trained on vgg with 9 layers, each task contains 5 categories; the y-axis denotes the indicator of ACC, FWT and BWT.