Continual learning of Single Shot Multibox Detector on Large dataset

Yicheng Ma
Department of Electrical and Computer Engineering
NYU Tandon School of Engineering
Brooklyn, NY 11201
ym1956@nyu.edu

Abstract

Recent works on continual learning have shown it is possible to alleviate catastrophic forgetting on multiple tasks across large datasets for two-stage object detectors. Inspired by the work of XiaLei et al[2] and Wang et al[5], this project proposed a regularization method to mitigate catastrophic forgetting for one of the famous one-stage object detector-Single Shot Multibox Detector or SSD. The basic idea of the proposed method is to regularize the performance of the current model feature extractors to force them extract similar features which are of great significance in the previous model. Experimental results on Pascal VOC datasets show that the regularization method effectively preserve the model performance of previous task(VOC 2007) after training on a new task(VOC 2012).

1 Introduction

Humans and animals are gifted with the ability to acquire, distill, transfer information and learn from it incrementally in their lifespan. This ability, often referred to as lifelong learning, or continual learning, is come from our complex neurocognitive mechanisms which also influence our sensorimotor skills and long-term memory consolidation. Apparently, the continual learning ability is crucial for our humans to survive the changing environments, as well as for autonomous agents such as robots and self-driving mobiles and computational learning systems, especially deep learning models. Although deep learning models have outperformed humans in many aspects including object recognition, video gaming and style imitation, the intrinsically simple design of neural networks doesn't simulate our neurocongnitive mechanisms, which leading to the catastrophic forgetting phenomenon, i.e., a phenomenon that occurs when training a model on a current task result in a rapid deterioration of the model's performance on previous tasks[4].

Retraining or joint learning can significantly mitigate the impact of catastrophic forgetting. However, in real-world cases, the full version legacy data of previous tasks are usually expensive to acquire or it can be lost, especially for the online learning setting. For example, a mobilenet SSD variant is designed to be deployed on mobile device, and the previous training data of this model maybe inaccessible for the user. Even if all the previous training data is accessible, the limited resources of the device cannot handle such a large volume of data.

Therefore, continual learning has been a long-standing challenge for deep learning models when the model is exposed to a continuous steam of data from different tasks where the number of tasks is not known in advance. Generally, for deep neural networks, there are three main approaches to mitigate catastrophic forgetting, which are[3]:

- Regularization: impose constraints on model parameters updating while keep the model architecture fixed, to preserve the information of previous tasks.

Preprint. Under review.

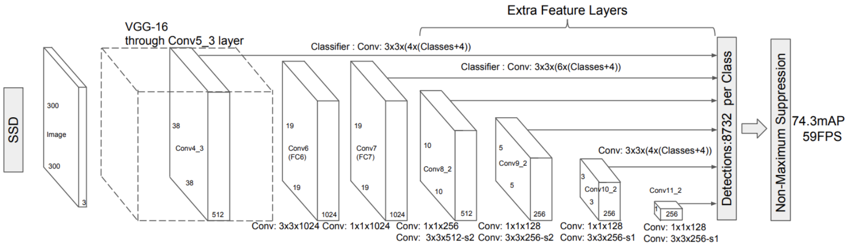

Figure 1: Architecture of SSD model with a VGG-16 back-bone[1].

- Dynamic architectures: dynamically expand the neural network with more neurons when new information incomes.
- Complementary learning systems and memory replay: directly store some examples of previous tasks and replay them or indirectly sample from a generative model to perform pseudo-rehearsal.

The proposed method of this project falls into the first category, and experimental results show that it is efficient to preserve the comparable performance of previous tasks without using more resources(extra neurons or memory buffers).

2 Project Approach

In the section, the accomplished parts will be discussed in details. Unaccomplished parts and future works will be discussed in Section 4.

2.1 Continual Learning Scenario

The planning two-task continual learning scenarios are: (i) both domains and categories are the same(use VOC 2007 as task A and VOC 2012 as task B); (ii) Same domain, different categories(use VOC 2012 with 20 categories as task A and COCO 2017 with 80 categories as task B). However, due to unfixed bugs of models trained on COCO 2017, only the first scenario is accomplished and experimented.

2.2 Framework

The object detection framework used in this project is Single Shot Multibox Detector(SSD)[1], which belongs to one-stage object detectors. A VGG-16 network is used as the back-bone. The architecture of the SSD model is shown in Figure 1,

Specifically, in the first scenario, for task A, a model A is trained on VOC 2007 trainval dataset using the standard object detection loss and is frozen after training; for task B, a new model B is trained on VOC 2012 train dataset using the proposed feature map loss (Section 2.3) with the frozen model A as a guidance. After learning on the task B, model A can be discarded and model B is used to do inference for both taks A and task B.

The loss function for task B is defined as:

$$L_{taskB} = L^B_{det} + L_{FML}$$

where L^B_{det} is the standard SSD detection loss and L_{FML} is the proposed feature map loss which mitigate the catastrophic forgetting.

2.3 Feature Map Loss

Similar to the Attentive Feature Distillation loss introduced in [2], the proposed feature map loss use the normalized self-attention maps as weights for each level's feature maps. Here, the self-attention

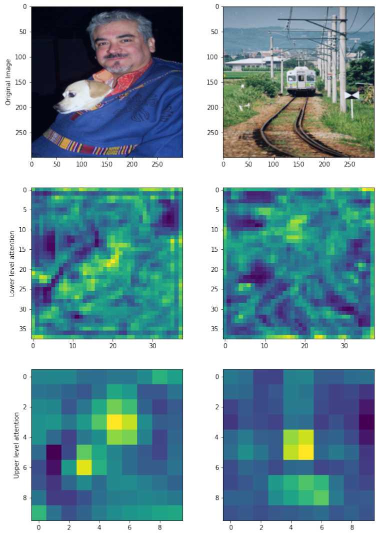

Figure 2: Example of attention maps. Original images(top row), lower level attention(middle row) and upper level attention(bottom row) from VOC 2007 dataset.

maps are defined as the summation across all the channels of the feature maps[2]. For each level's feature maps, an element-wise multiplication of the corresponding self-attention map is broadcasted to all channels to distill the important context information. Since SSD has a different structure of Faster-RCNN, it has multi-scale feature maps but doesn't contain RPN to generate region proposals, the proposed feature map loss use the lower level self-attention maps(outputs of conv4_3 and conv7) to extract fine-grained local attention information and use the upper level self-attention maps(outputs of conv8_2 to conv11_2) to extract global attention information, as shown in Figure 2. Different trade-off parameters are applied to different level feature maps to balance the influence of local attention and global attention.

The feature map loss for SSD is defined as:

$$L_{FML}(F^B) = \sum_i^N \frac{\lambda_i}{2} \| A_i^B \odot F_i^B(\mathbf{x}_j) - A_i^A \odot F_i^A(\mathbf{x}_j) \|^2$$

where $\odot$ indicates element-wise multiplication and is broadcasted to all channels of F, F_i^B and F_i^A are the feature maps of the current model B and the frozen previous model A, i indicates different levels and A_i is the corresponding self-attention map for level i, which is defined as:

$$A_i = \frac{\sum_{k=1}^C F_{i,k}(\mathbf{x}_j)}{\| \sum_{k=1}^C F_{i,k}(\mathbf{x}_j) \|}$$

3 Experiments

Only conduct experiments on the first scenario. The MS COCO experiments didn't give valid results due to unfixed bugs.

3.1 Working environment

The experiment environment is built on the operating system Ubuntu 20.04 with python version 3.8.5. The GPU model is Gefore RTX 2080ti with 11GB RAM. The Nvidia driver version is 450.119. Pytorch version 1.6.0 is installed with corresponding CUDA version 10.1 and cuDNN version 8.0.

Table 1: Analysis of model performance(mAP) after training on task B. Arrows indicate order of learning.

Tasks	VOC 2007→	VOC 2012
Joint learning	0.739	0.701
Standard Loss(Only 2007)	0.712	-
Standard Loss(Only 2012)	0.699	0.670
Feature Map Loss	0.713	0.675

3.2 Dataset

For the first scenario, Pascal VOC 2007 trainval dataset is used as the training set for task A, Pascal VOC 2012 train dataset is used as the training set for taskB. VOC 2007 test dataset and VOC 2012 valiadation dataset are used for evaluation. For the second scenario, VOC 2012 is planned to be task A and COCO 2017 is planned to be task B.

3.3 Training details

Following the settings of [1], the VGG-based SSD models are trained with 8732 priors for 120k iterations. The learning rate is set to 1e-3 at first, and it decays by 0.1 after 80k and 100k iterations. The batch size is set to 8. For task B, λ_i of lower levels are set to 1e-4 and λ_i of upper levels are set to 1e-3.

3.4 Results

The standard Pascal VOC mean average precision(mAP) is used for evaluation. The joint training model's performance is used as the upper bound of the proposed approach. The frozen model A and a model trained with standard SSD detection loss on VOC 2012 are used as comparison to analyze forgetting.

The experiment results are shown in Table 1. From the table, it can be observed that without the proposed feature map loss, after training on task B, the model forgets the information of task A, which leads to 0.013 mAP drops on VOC 2007 test dataset. However, if use the proposed approach, after training on task B, the model still has a comparable performance on previous task(in this case, even outperforms previous model) without degenerating the performance on current task(even has 0.005 mAP improvement). Since both the task A and task B are in the same domain with the same categories, the improvement of performance using proposed method and joint learning is reasonable and the performance of joint learning can always be the upper bound of continual learning algorithm under this circumstance.

4 Conclusion

In this project, a proposed feature map loss regularization method is investigated on a two-task continual learning scenario. The experimental result shows that this method is valid for tasks in the same domain with the same categories. However, further experiments on different domain and especially different categories tasks are necessary to verify the effectiveness of this method. The planned experiment includes a scenario of tasks in the same domain but have different categories(VOC 2012 and COCO 2017). Due to the unfixed bugs of training models on COCO dataset(localization loss didn't converge leading to abnormal bounding boxes and extremely low mAP), the experiment on a more complex scenario is impeded. Future works will focus on fixing these bugs and do more experiments on more continual learning scenarios.

References

[1] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[2] X. Liu, H. Yang, A. Ravichandran, R. Bhotika, and S. Soatto. Multi-task incremental learning for object detection. *arXiv e-prints*, pages arXiv–2002, 2020.

[3] G. I. Parisi, R. Kemker, J. L. Part, C. Kanan, and S. Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

[4] Y. Teng, A. Choromanska, and M. Campbell. Continual learning with direction-constrained optimization. *arXiv preprint arXiv:2011.12581*, 2020.

[5] W. Zhou, S. Chang, N. Sosa, H. Hamann, and D. Cox. Lifelong object detection. *arXiv preprint arXiv:2009.01129*, 2020.