LaTeX Author Guidelines for CVPR Proceedings

Anonymous CVPR submission

Paper ID *****

Abstract

The continual learning is a meaningful and challenging task in deep learning and computer vision, which is a crucial step to make artificial general intelligence come true. However, the notorious catastrophic forgetting hinders the advancement and development of this learning paradigm. In this work, we propose a relatively effective training framework to alleviate forgetting problem by jointly learning explicit knowledge and implicit knowledge. We use replay memory module to dynamically store the sparse explicit from the representative images and adopt a better backbone to contain rich implicit knowledge about previous classes. By means of this strategy in the form of supervised learning, we surpass the majority of rivals and achieve top performance in 3rd CLVision competition.

1. Introduction

Incremental object detection requires a detector to continually learn to recognize the objects from classes that never show up in the previous train phases on condition that this detector preserves the cognition to previous classes throughout the training data flow. Unfortunately, if we do not take any measurement to interfere the process of training, the deep learning based detector has a poor performance on the old classes. This phenomenon is named as catastrophic forgetting [2]. To conquer this problem, numerous methods have been presented in recent years. Despite of improvement in this research field, catastrophic problem still is still a long-standing challenge.

The prevalent works design different models to alleviate the forgetting. Basically, they can be roughly divided into three categories, i.e., meta learning based methods [5], distillation based methods [3]. And replay [6] as an simple and effective trick usually is adopted in these two different branches. After the experimental validation, we take the advantages of replay to design our framework.

In specific, we both leverage the explicit and implicit knowledge to jointly train our object detector. More specifically, We maintain a memory with fixed size to dynamically store the representative images from the old classes. We keep the stored images diverse to make them represent the corresponding class data. The stored images will be trained with novel classes at current train phase. However, the memory is limited and not able to contain all information contained in old classes.

The fitting capability of model is also a important factor to address the catastrophic forgetting. While the scale of the whole data is fixed, the more knowledge model can learned from train data, the better performance model can achieved to remember the old classes. Specifically, due to the remarkable performance of transformer [8] on the large scale data, we pick a transformer-based model as our start-up model which is pre-trained on the Image1K.

Equipped with such train strategy and model, we ranked at a third position in the continual category-level object detection track of the 3rd CLVision challenge.

2. Method

The challenge allows us to use no more than 70M total model parameters. And allows us to use a maximum number of caches of 5000. Based on this, we chose a model with very good performance and equipped with an adaptive replay buffer.

2.1. Model

Transformer-based models have recently achieved very good results on many vision domain tasks. Compared with CNN-based models, the attention mechanism of transformer can find relevant information over the whole feature graph, thus making the extracted features more semantically informative. Therefore, we choose the transformer-based backbone to extract data features for downstream detection task.

The larger learning capacity of the model, the more likely it perform better on incremental learning tasks. We use as many parameters as possible to ensure that the model can learn a large amount of knowledge. Swin-transformer [4], a hierarchical transformer whose representation is computed with shifted windows, whose recent excellent performance on classification tasks is evidence of its very good

Model	Replay	AvgAP
Faster-RCNN+ResNet50	✓	0.3881
Faster-RCNN+SwinTiny	✓	0.3879
Faster-RCNN+SwinSmall	✓	**0.4240**

Table 1. Results of model comparison. The AvgAp means the score of our submission. SwinSmall performs best.

feature extraction ability. We use it to act as a backbone for the detector, and after comparison, we chose the backbone model named swin-small.

Generally, the two-stage detector tends to perform better than the one-stage detector. Faster-RCNN, [7] a classical detector structure, still performs well on many detection tasks. The Cascade-RCNN, proposed by [1], which uses different thresholds at different stages, makes the model faster and more accurate when learning. We prefer to use Cascade RCNN, but due to the lack of implementation, we finally choose Faster-Rcnn to act as our detector architecture.

2.2. Replay Buffer

One of the main factors of catastrophic forgetting is the inability to access historical data. Cache tends to be very large and take up linearly more memory over time, but it is a very effective solution to catastrophic forgetting. Therefore, if it is possible to have a cache to keep historical data, then the model can perform well both on new classes and old classes after a new experience.

We were allowed to use a replay buffer with a maximum number of 5000, so we kept adding data from new experiences and reducing data from old experiences after each experience round, while keeping the total number constant, this allows us to make the best use of the replay buffer.

3. Experiment

3.1. Setting

We set the training batchsize to 1 because the reference server has a graphics memory of 16 G. We set the number of epochs of the training phase to 5 because the maximum solution time of the category detection track is 24 hours. We use SGD optimizer because it performs better in most cases. Also we used the warmup strategy for the learning rate. For the replay buffer, we use a replay plugin with a maximum size of 5000. After each experience round, a certain number of samples are randomly selected in the experience, and the number of samples retained in each experience round is continuously adjusted so that the sum of the number is a fixed 5000.

3.2. Model Comparison

We choose Faster-RCNN as our final two-stage detector framework. With both using the replay plugin, we replaced Faster-RCNN's feature extraction backbone and tried three models: resnet50, swintiny, and swinsamll. As shown in Tab. 1, the results of resnet50 and swintiny are relatively similar, compared with them, swinsmall is significantly better, and finally we chose swinsmall to act as backbone. We got the third place in the classification detection track.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 2

[2] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999. 1

[3] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 1

[4] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 1

[5] Juan-Manuel Perez-Rua, Xiatian Zhu, Timothy M Hospedales, and Tao Xiang. Incremental few-shot object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13846–13855, 2020. 1

[6] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 1

[7] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 2

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 1