Intelligent Replay Sampling for Lifelong Object Recognition

Vidit Goel[1], Debdoot Sheet[2], Somesh Kumar[3]

***Abstract*— In this work we aim to solve the image recognition task in an incremental manner. Lifelong learning is known to suffer from catastrophic forgetting. Various methods such as regulatory, architecture and replay based methods have been proposed to to solve the problem. We propose a novel, simple and efficient method for sampling data from the buffer and using it as replay data. It uses the best accuracy that can be achieve on a batch of data and the current accuracy on the same batch to judge the importance of the batch. This work was done as a part of Lifelong Object Recognition Challenge, IROS, 2019. We achieved an accuracy of 97.04% on the challenge dataset and were ranked 4^{th}.**

I. INTRODUCTION

With the advent of Convolutional Neural Networks (CNNs) there are many solution to object recognition surpassing human level accuracy[1], [2], [3]. But all the above solutions are not practical as they can perform only on the data that they have been trained on. The performance will significantly drop if the domain of the data is changed such as angle at which it is seen or the illumination condition or the surroundings(occlusion, clutter) [4]. To address these issues we have to learn the tasks in an incremental way i.e Lifelong learning. All the continual learning strategies can be broadly classified into three categories as described in[5]

1) **Regularization Approaches** These methods deal with restricting the change in weights which are important for a particular task based on some metric. For example elastic weight consolidation method[6] calculate fisher information matrix and penalizes if there is change in weights which are important to particular task. As a drawback it is computationally expensive to calculate fisher matrix. [7] proposed an efficient way to calculate importance of weight. Another work by Rahaf et al.[8] restricts the number of parameters that can be learned in a task.
2) **Dynamic Architectures** In this, architectures are modified depending upon the task. Methods that can be classified in this category are [9], [10]. There are some methods which combine both regularization and architectural strategies such as CWR and AR1[11].
3) **Replay Based Methods** Unlike the above methods in this method certain amount of data is stored from previous task which is replayed with the current task to avoid catastrophic forgetting. Gradient Episodic Memory was an elegant solution proposed in [12] but this method requires a lot more memory compared to other methods hence it is not feasible to use this method. FearNet[13] incremental class learning is inspired by studies of recall and consolidation in the mammalian brain.

[1]Vidit Goel is with the Department of Electrical Engineering, Indian Institute of Technology, Kharagpur, West Bengal 721302, India. `gvidit98@gmail.com`

[2]Debdoot Sheet is with the Department of Electrical Engineering, Indian Institute of Technology, Kharagpur, West Bengal 721302, India. `debdoot@ee.iitkgp.ac.in`

[3]Somesh Kumar is with the Department of Mathematics, Indian Institute of Technology, Kharagpur, West Bengal 721302, India. `smsh@maths.iitkgp.ac.in`

The solution proposed by us will fall in replay based methods. We show how to intelligently sample elements for replay to reduce the replay memory size significantly.

In this work we particularly target new instance[11] continual learning, where the number of classes of object remain same in different tasks but the distribution of data shifts gradually due changes in illumination, surroundings(occlusion, clutter) and angle of camera.

II. METHOD

A. Problem Definition

Lets start by defining the problem statement. Suppose there are N tasks, in each task given an image we have to classify it as an object. We have to train the network in a sequential manner i.e. when training for task t_n we don't have access to training data of task t_i where $i < n$. We do have access to validation data for previous tasks. Finally, the network trained on N^{th} will be tested on all the previous task.

B. Intelligent Replay Sampling

CNNs map the input data(images) to some latent space. They do so by extracting high dimensional features. The features of all the images belonging to a particular object should be near by in the space and that of different object should be far away. All the features belonging to an object form a distribution in the high dimensional space. This distribution is dependent on training data itself.

For the scenario described above there is a gradual shift in the distribution of training data across tasks. Suppose we train a network on a task t_n and it learns some feature representation of the images in the task, now when we train on the task t_{n+1} it learns the feature representation for images in task t_{n+1}, but as the distribution of data is task t_{n+1} is different, accuracy drops for images in task t_n.

This approach samples validation data from the buffer and use it as replay data. It intelligently creates the replay memory for a task. The replay memory will be such that it is an efficient representation of previous tasks data whose information is lost. The replay data is sampled from the validation of all the previous tasks. They train the network on task t_n and save the accuracy of batches of validation data. Next, when they train on task t_i $(i > n)$, they calculate the accuracy of same batches of validation data of task t_n. Then they store the top k batches from validation data of task t_n whose accuracy has dropped the most. This is done for all the tasks t_0 to t_{i-1}. Now when they train for task t_{i+1} they will combine the replay data and training data to train for the particular task.

```
Algorithm 1: Replay Data Sampling
Result: Replay_Data
initialization $F_i, val\_data_i, t_n, acc[], best\_acc[], topk$;
while While data in $val\_data_i$ do
    prec = Accuracy($F_i(data)$);
    ADD prec to $acc[]$;
end
if $i == n$ then
    ADD $acc$ to $best\_acc$;
else
    diff = $best\_acc - acc$ ;
    sort_diff = sort(diff);
    ADD $topk$ elements corresponding to sort_diff from
      $val\_data_i$ to Replay_Data
end
```

C. Training Details

Dataset used was OpenLORIS-Object dataset[14]. It is divided into 12 tasks, in each tasks we need to classify images to one of the objects from 69 classes. Each task has different settings like illumination, clutter, occlusion etc. Training set consisted of 14k images per task and validation consisted of 2k images per task. We used MobileNetV2[15] as object classifier with weight decay 2e-4, batch size of 16 and SGD optimizer. Image was resized to 224x224 before feeding it into network. Top 40 batches(640 images) were stored from each validation task. Pytorch was used for implementation. We start with learning 0.01 and decay it with a factor of 0.98 after every epoch. After the first epoch of next task learning rate is reset to 0.01. Another important thing to note is that when we sample replay data after training on task t_n we will be able to get data from tasks t_0 to t_{n-1}. So when we train on task t_{n+1}, accuracy will drop for task t_n. To handle this we generate the replay data for task t_n after training for 1 epoch on t_{n+1}. We want learning rate to be low when training first epoch of task t_{n+1} so that the accuracy of task t_n does not fall catastrophically. Therefore, the learning rate is reset to 0.01 after the first epoch. Each task was trained for 20 epochs except for task 11. Task 11 was trained for 30 epochs. We have made the code available.

D. Results

After training on all the tasks we achieved an accuracy 97.05%. Accuracy after training on different tasks in shown in Fig 1. Final accuracy is shown in table 1.

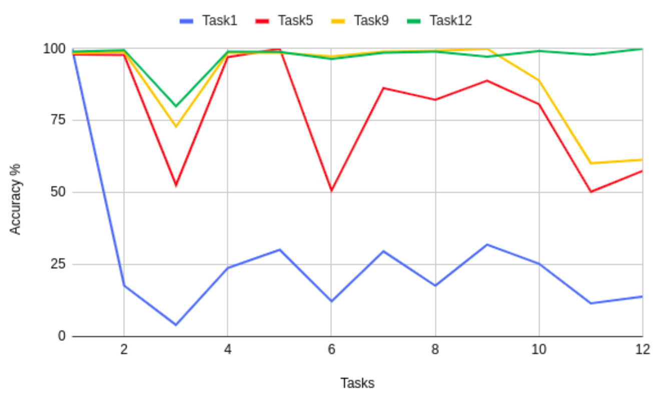

Fig. 1. Validation accuracy on all tasks after training on task 1, 5, 9, 12.

task	1	2	3	4	5	6
acc	98.94	99.52	80.04	98.94	98.85	96.44
task	7	8	9	10	11	12
acc	98.56	98.94	97.27	99.16	97.95	100.00

TABLE I

ACCURACY ON ALL THE TASKS AFTER TRAINING FOR TASK 12

REFERENCES

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
[2] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.
[3] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy. *arXiv preprint arXiv:1906.06423*, 2019.
[4] F. Feng, R. H. M. Chan, X. Shi, Y. Zhang, and Q. She. Challenges in task incremental learning for assistive robotics. *IEEE Access*, 8:3434–3441, 2020.
[5] German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural Networks*, 2019.
[6] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.
[7] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3987–3995. JMLR. org, 2017.
[8] Rahaf Aljundi, Marcus Rohrbach, and Tinne Tuytelaars. Selfless sequential learning. *arXiv preprint arXiv:1806.05421*, 2018.
[9] German I Parisi, Jun Tani, Cornelius Weber, and Stefan Wermter. Lifelong learning of human actions with deep neural network self-organization. *Neural Networks*, 96:137–149, 2017.
[10] German Ignacio Parisi, Jun Tani, Cornelius Weber, and Stefan Wermter. Lifelong learning of spatiotemporal representations with dual-memory recurrent self-organization. *Frontiers in neurorobotics*, 12:78, 2018.
[11] Davide Maltoni and Vincenzo Lomonaco. Continuous learning in single-incremental-task scenarios. *Neural Networks*, 116:56–73, 2019.
[12] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017.
[13] Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. *arXiv preprint arXiv:1711.10563*, 2017.

[14] Qi She, Fan Feng, Xinyue Hao, Qihan Yang, Chuanlin Lan, Vincenzo Lomonaco, Xuesong Shi, Zhengwei Wang, Yao Guo, Yimin Zhang, et al. Openloris-object: A robotic vision dataset and benchmark for lifelong deep learning. *arXiv preprint arXiv:1911.06487*, 2019.

[15] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018.