Projection as Pareto Improvement: Multi-objective optimization for Multi-label Continual Learning

Yingru Li, Xiang Liu
The Chinese University of Hong Kong, Shenzhen, China
yingruli@link.cuhk.edu.cn

Abstract

Modern deep learning techniques are facing the *catastrophic forgetting* problem that when the deep network is trained on a new task without seeing the data in the old task, knowledge learned from the old task would be easily forgotten, i.e., test accuracy in old task drops tremendously.

To overcome catastrophic forgetting in a sequence of multi-label classification tasks, it requires multi-label continual learning. We first rethink continual learning problem in the framework of *multi-objective optimization*. Then, we develop the *Projection as Pareto Improvement* (PPI) method within the framework. PPI is a strategy for updating the hidden layer which could locally guarantee better performance on the new task and no-worse performance on the old tasks. For the case that only task-specific ground truth labels and data are available in the new task, we adopt the model distillation strategy to mimic pseudo soft labels that are related to previous tasks. We also developed a benchmark for the multi-label continual learning problem based on Pascal VOC from which we hope further research will benefit. The extensive empirical studies show the supremacy of our approaches.

1 Introduction

In modern deep learning filed, one of the most common problems is that the testing accuracy of an old task declines while training a new task, and the knowledge that learned in the old tasks would be forgotten, which indicates the catastrophic forgetting problem [3, 7, 11]. This problem is caused mainly by data distributional shift and is one of the major obstacles on the road for deep network based artificial agents to achieve the human-level intelligence.

Recent works show promising progress in tackling the continual learning research by two main strategies. One strategy, retrospection, is to exploit a certain amount of previous tasks' information by external memory mechanism to store few real samples from old tasks [13, 10], by a generative model to memorize sample distribution of all past tasks [16, 5], or by model distillation to memorize marginal distribution [9]. Another strategy is to reduce the representational overlap between tasks by structural regularization [7, 1, 17, 15, 12, 8], that seeks to prevent major changes in the important weights w.r.t. past tasks when training and optimizing the new task.

However, almost all these works only considered the single-label multi-class classification problem but ignored the multi-label multi-class classification problem. Directly adapting the well-established single-label continual learning approaches to solve its multi-label variants is not applicable. For example, learning without forgetting (LwF) [9] requires pseudo soft-targets from the old model, that have potential conflicts with ground truth multi-label, as a supervision signal within the training procedure; Gradient Episodic Memory (GEM) [10] requires the data sample in old task as the memory for training process, which is not available in many real-world scenarios.

Paper Report of Final Project for CIE6032 (MDS6232)- Selected Topics in Deep Learning.

To overcome catastrophic forgetting in a sequence of incremental multi-label classification tasks, it requires multi-label continual learning. Inspired by MTLasMOO [14], We first rethink continual learning problem in the framework of multi-objective optimization. We then develop the Projection as Pareto Improvement (PPI) method within the framework. PPI is a strategy for updating the hidden layer which could locally guarantee better performance on the new task and no-worse performance on the old tasks. Our framework focus on the hard case that the algorithms cannot access data from old tasks when facing the new task. Combined with model distillation techniques, our framework can even handle harder case that only task-specific data and labels are available for training each task. We also developed a benchmark for the multi-label continual learning problem that contained three tasks based on Pascal VOC. To the best of our knowledge, we are the first to tackle multi-label continual learning problem within deep learning systems generally.

2 The Proposed Algorithm

2.1 Continual Learning as Multi-Objective Optimization

Consider a continual learning (CL) problem over a collection of task spaces $\{\mathcal{X}^t \times \mathcal{Y}^t\}_{t\in[T]}$ and fixed joint probability distribution P^t over the t^{th} task space, such that a sequence of task-specific dataset arrives by the order of tasks (from task 1 to task T)

$$\{\mathbf{x}_i^1, y_i^1\}_{i\in[N^1]}, \cdots, \{\mathbf{x}_i^t, y_i^t\}_{i\in[N^t]}, \cdots, \{\mathbf{x}_i^T, y_i^T\}_{i\in[N^T]} \tag{1}$$

where T is the number of tasks, N^t is the number of data points in task t, $N = \sum_t N^t$ is the total number of data points, and y_i^t is the label for the i^{th} data point $\mathbf{x}_i^t$ of the t^{th} task satisfying $(\mathbf{x}_i^t, y_i^t) \sim P^t$. [1] We further consider a parametric hypothesis class per task as $f^t(\mathbf{x}; \boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^t) : \mathcal{X} \rightarrow \mathcal{Y}^t$ that is accessible for all input $\mathbf{x} \in \mathcal{X} \triangleq \bigcup_t \mathcal{X}^t$, such that some parameters ($\boldsymbol{\theta}^{sh}$) are shared between tasks and some ($\boldsymbol{\theta}^t$) are task-specific within the whole parameters $\boldsymbol{\theta}$. We also consider task-specific loss functions $\mathcal{L}^t(\cdot,\cdot) : \mathcal{Y}^t \times \mathcal{Y}^t \rightarrow \mathbb{R}^+$ and then the task-specific empirical loss $\hat{\mathcal{L}}^t(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^t) \triangleq \frac{1}{N^t}\sum_i \mathcal{L}^t\big(f^t(\mathbf{x}_i^t; \boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^t), y_i^t\big)$.

The continual learning procedure is basically in the following pattern. With the sequence of task-specific dataset arriving, when we enter the task t, we can only access to t^{th} dataset $\{\mathbf{x}_i^t, y_i^t\}_{i\in[N^t]}$ and parameters of previous best model $\boldsymbol{\theta}^*(t-1)$, discarding all previous datasets, and then train the current dataset to get the new model

$$\boldsymbol{\theta}^*(t) = \operatorname*{arg\,min}_{\substack{\boldsymbol{\theta}^{sh}, \\ \boldsymbol{\theta}^1,\ldots,\boldsymbol{\theta}^T}} \quad \hat{\mathcal{L}}^t(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^t | \boldsymbol{\theta}^*(t-1)) + \mathcal{R}(\mathbf{x}^t, \boldsymbol{\theta} | \boldsymbol{\theta}^*(t-1)) \tag{2}$$

where $\boldsymbol{\theta}^*(0)$ is the random or pretrained initialization of the parameters and $\mathcal{R}(\cdot)$ is some regularization. Then we define the offline optimal as

$$\boldsymbol{\theta}^* = \operatorname*{arg\,min}_{\substack{\boldsymbol{\theta}^{sh}, \\ \boldsymbol{\theta}^1,\ldots,\boldsymbol{\theta}^T}} \quad \sum_{t=1}^{T} c^t \hat{\mathcal{L}}^t(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^t) \tag{3}$$

where c^t can be $\frac{N^t}{N}$ or other static or dynamic weights associated to task t. Our ultimate goal is to guarantee the performance of the final model $\boldsymbol{\theta}^*(T)$ after seeing all data. One possible metric is regret-like function:

$$\sum_{t=1}^{T} c^t \hat{\mathcal{L}}^t(\boldsymbol{\theta}^{sh^*}(T), \boldsymbol{\theta}^{t^*}(T)) - \sum_{t=1}^{T} c^t \hat{\mathcal{L}}^t(\boldsymbol{\theta}^{sh^*}, \boldsymbol{\theta}^{t^*}) \tag{4}$$

If the labels of each task-specific dataset can be augmented with ground truth labels in other tasks, we could have the augmented dataset $\{\mathbf{x}_i, y_i^1, \cdots, y_i^T\}_{i\in[N]}$.

We then define the task-unaware empirical loss $\bar{\mathcal{L}}(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^t) \triangleq \frac{1}{N}\sum_i \sum_t \mathcal{L}^t\big(f^t(\mathbf{x}_i; \boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^t), y_i^t\big)$ and we can have another regret-like metric:

$$\bar{\mathcal{L}}(\boldsymbol{\theta}^{sh^*}(T), \boldsymbol{\theta}^{t^*}(T)) - \bar{\mathcal{L}}(\boldsymbol{\theta}^{sh^*}, \boldsymbol{\theta}^{t^*}) \tag{5}$$

[1] In our multi-label continual learning problem setting, the task-specific label sets $\{\mathcal{Y}^t\}_{t\in[T]}$ are mutually exclusive.

2.2 Model distillation as data label augmentation

When entering the task t, the label of task-specific input $\mathbf{x}^t$ can be augmented using previous best model

$$\tilde{y}^{t'|t} = f^{t'}(\mathbf{x}^t; \boldsymbol{\theta}^{sh^*}(t-1), \boldsymbol{\theta}^{t'^*}(t-1)), \quad \forall t' < t \tag{6}$$

where the pseudo-soft-label $\tilde{y}^{t'|t}$ is probability distribution over label set $\mathcal{Y}^{t'}$ generated from $f^{t'}$ on data input $\mathbf{x}^t$. Then we have augmented task-specific dataset $\{\mathbf{x}_i^t, \tilde{y}_i^{1|t}, \cdots, \tilde{y}_i^{t-1|t}, y_i^t\}_{i\in[N^t]}$.

Assume binary cross entropy loss function in our multi-label continual learning setting, then we can define $\tilde{\mathcal{L}}^t(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^t) \triangleq \hat{\mathcal{L}}^t(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^t)$ and for all $t' < t$, denote

$$\tilde{\mathcal{L}}^{t'}(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^{t'}) \triangleq \frac{1}{N^{t'}} \sum_i \mathcal{L}^{t'}\left(f^{t'}(\mathbf{x}_i^t; \boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^{t'}), \tilde{y}_i^{t'|t}\right) \tag{7}$$

At current stage t, one approach to solve the multi-label continual learning problem is

$$\begin{aligned}
\boldsymbol{\theta}^*(t) &= \underset{\substack{\boldsymbol{\theta}^{sh},\\ \boldsymbol{\theta}^1,\ldots,\boldsymbol{\theta}^T}}{\arg\min} \quad \sum_{t'=1}^{t} c^{t'} \tilde{\mathcal{L}}^{t'}(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^{t'} | \boldsymbol{\theta}^*(t-1)) && (8)\\
&= \underset{\substack{\boldsymbol{\theta}^{sh},\\ \boldsymbol{\theta}^1,\ldots,\boldsymbol{\theta}^T}}{\arg\min} \quad \tilde{\mathcal{L}}^t(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^t | \boldsymbol{\theta}^*(t-1)) + \sum_{t'=1}^{t-1} \frac{c^{t'}}{c^t} \tilde{\mathcal{L}}^{t'}(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^{t'} | \boldsymbol{\theta}^*(t-1)) && (9)\\
&\triangleq \underset{\substack{\boldsymbol{\theta}^{sh},\\ \boldsymbol{\theta}^1,\ldots,\boldsymbol{\theta}^T}}{\arg\min} \quad \hat{\mathcal{L}}^t(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^t | \boldsymbol{\theta}^*(t-1)) + \mathcal{R}(\mathbf{x}^t, \boldsymbol{\theta} | \boldsymbol{\theta}^*(t-1)) && (10)
\end{aligned}$$

This is an approximation of to the offline problem (3) when we can only access task-specific input in task t.

Although the weighted summation formulation (8) is intuitively appealing, it typically either requires an expensive grid search over various scalings or the use of a heuristic [6, 2]. A basic justification for scaling is that it is not possible to define global optimality in the continual learning setting. Consider two sets of solutions $\boldsymbol{\theta}$ and $\bar{\boldsymbol{\theta}}$ such that $\tilde{\mathcal{L}}^{t_1}(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^{t_1}) < \tilde{\mathcal{L}}^{t_1}(\bar{\boldsymbol{\theta}}^{sh}, \bar{\boldsymbol{\theta}}^{t_1})$ and $\tilde{\mathcal{L}}^{t_2}(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^{t_2}) > \tilde{\mathcal{L}}^{t_2}(\bar{\boldsymbol{\theta}}^{sh}, \bar{\boldsymbol{\theta}}^{t_2})$, for some tasks t_1 and t_2. In other words, solution $\boldsymbol{\theta}$ is better for task t_1 whereas $\bar{\boldsymbol{\theta}}$ is better for t_2. It is not possible to compare these two solutions without a pairwise importance of tasks, which is typically not available.

Alternatively, multi-lable continual learning at each stage can be formulated as multi-objective optimization: optimizing a collection of possibly conflicting objectives. Inspired by MTLasMOO [14], we specify the multi-objective optimization formulation of MCL at satge t using a vector-valued loss $\mathbf{L}$:

$$\min_{\substack{\boldsymbol{\theta}^{sh},\\ \boldsymbol{\theta}^1,\ldots,\boldsymbol{\theta}^T}} \mathbf{L}(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^T) = \min_{\substack{\boldsymbol{\theta}^{sh},\\ \boldsymbol{\theta}^1,\ldots,\boldsymbol{\theta}^T}} \left(\tilde{\mathcal{L}}^1(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^1), \ldots, \tilde{\mathcal{L}}^t(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^t)\right)^{\intercal}. \tag{11}$$

The goal of multi-objective optimization is achieving Pareto optimality.

Definition 1 (Pareto optimality for MCL at stage t)**.**

(a) *A solution $\boldsymbol{\theta}$ dominates a solution $\bar{\boldsymbol{\theta}}$ if $\tilde{\mathcal{L}}^{t'}(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^{t'}) \leq \tilde{\mathcal{L}}^{t'}(\bar{\boldsymbol{\theta}}^{sh}, \bar{\boldsymbol{\theta}}^{t'})$ for all tasks $t' \in [t]$ and $\mathbf{L}(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^T) \neq \mathbf{L}(\bar{\boldsymbol{\theta}}^{sh}, \bar{\boldsymbol{\theta}}^1, \ldots, \bar{\boldsymbol{\theta}}^T)$.*

(b) *A solution $\boldsymbol{\theta}^\star$ is called Pareto optimal if there exists no solution $\boldsymbol{\theta}$ that dominates $\boldsymbol{\theta}^\star$.*

The set of Pareto optimal solutions is called the Pareto set ($\mathcal{P}_{\boldsymbol{\theta}}$) and its image is called the Pareto front ($\mathcal{P}_{\mathbf{L}} = \{\mathbf{L}(\boldsymbol{\theta})\}_{\boldsymbol{\theta}\in\mathcal{P}_{\boldsymbol{\theta}}}$).

Definition 2 (Pareto Improvement for MCL at stage t)**.**

(a) *A Pareto improvement is the change of solutions $\Delta\boldsymbol{\theta} = \boldsymbol{\theta} - \bar{\boldsymbol{\theta}}$ that makes at least one loss function lower without making any other loss function larger, i.e. the solution $\boldsymbol{\theta}$ dominates a solution $\bar{\boldsymbol{\theta}}$.*

(b) *Assume $\boldsymbol{\theta}^*(t-1)$ is the solution (lies in the Pareto set) at stage $t-1$. Ideally solving catastrophic forgetting problem in continual learning at stage t is to find a solution $\boldsymbol{\theta}^*(t)$ which minimizes the t-th task-specific loss function and also guarantee $(\boldsymbol{\theta}^*(t) - \boldsymbol{\theta}^*(t-1))$ is a Pareto improvement with respect to $\mathbf{L}$.*

2.3 Solving the optimization problem

We first present the updating rule based on block coordinate descent. We conjecture this framework can converge to Pareto stationary point. To solve the direction finding sub-problem, we can do simple weighted average in algorithm (3) or using our proposed PPI method in algorithm (2). PPI seeks to locally reduce the loss associated to task t while at the same time not increasing the loss associated to precious tasks. We conjecture local Pareto improvement can lead to (approximate) Pareto optimality.

Algorithm 1 Update Equations for MCL at stage t

```
while Not reach terminate condition do
    for $t' = 1$ to $t$ do
        $\boldsymbol{\theta}^{t'} = \boldsymbol{\theta}^{t'} - \eta \nabla_{\boldsymbol{\theta}^{t'}} \tilde{\mathcal{L}}^{t'}(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^{t'})$        ▷ Gradient descent on task-specific parameters
    end for
end while
while Not reach terminate condition do
    $\tilde{g}$ = DIRECTIONFINDING $(\boldsymbol{\theta}, t)$        ▷ find a common descent direction with preference
    $\boldsymbol{\theta}^{sh} = \boldsymbol{\theta}^{sh} - \eta \tilde{g}$        ▷ Gradient descent on shared parameters
end while
```

Algorithm 2 DirectionFinding Subproblem: Projection as local Pareto Improvement

```
procedure PROJECTION($\boldsymbol{\theta}, t$)
    for $t' = 1$ to $t$ do
        $g^{t'} = \nabla_{\boldsymbol{\theta}^{sh}} \tilde{\mathcal{L}}^{t'}(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^{t'})$
    end for
    $\tilde{g} = \arg\min_{\tilde{g}} \frac{1}{2} \|\tilde{g} - g^t\|^2$,
    s.t. $\langle \tilde{g}, g^{t'} \rangle = \langle \tilde{g}, \nabla_{\boldsymbol{\theta}^{sh}} \tilde{\mathcal{L}}^{t'}(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^{t'}) \rangle \geq 0, \quad \forall t' < t$
    ▷ This quadratic programming problem can be efficiently solved by solving its dual problem.
    return $\tilde{g}$
end procedure
```

Algorithm 3 DirectionFinding Subproblem: Simple average

```
procedure AVERAGE($\boldsymbol{\theta}, t$)
    Initialize $\boldsymbol{\alpha} = (\alpha^1, \ldots, \alpha^t) = (\frac{1}{t}, \ldots, \frac{1}{t})$
    Find good weights $\boldsymbol{\alpha}$
    $\tilde{g} = \sum_{t'=1}^{t} \alpha^{t'} \nabla_{\boldsymbol{\theta}^{sh}} \tilde{\mathcal{L}}^{t'}(\boldsymbol{\theta}^{sh}, \boldsymbol{\theta}^{t'})$        ▷ Weighted average on each task-specific loss function.
    return $\tilde{g}$
end procedure
```

3 Dataset Description

The trainval dataset of Pascal VOC 2012 (11540 images with 27450 labeled items from 20 classes), and both the trainval and the test dataset of Pascal VOC 2007 (9963 images with 24640 labeled items from 20 classes) are used in our paper. The number of positive samples for each class of the whole dataset is shown in the Figure 1.

Three subsets of images from the original whole Pascal VOC Dataset are created to mimic a continual learning task in real life. The first one contains the images which only have positive labels for the first 7 classes. The second subset contains the images that must have positive labels for the second 7 classes and may contain positive labels for the first 7 classes. The last subset is consisted of the

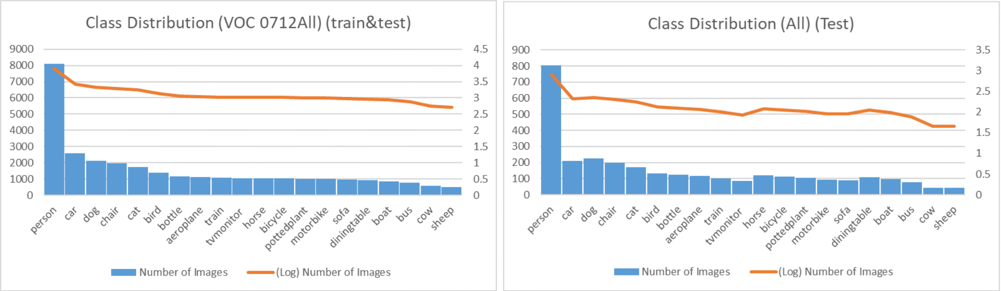
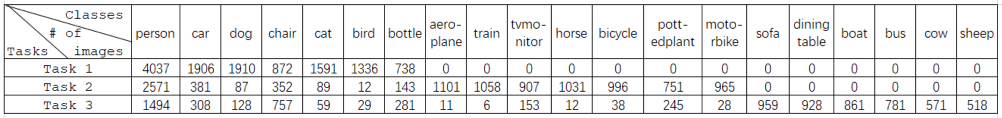


Figure 1: Class distribution for the whole dataset and testing set

images that must have positive labels for the last 6 classes and may have positive labels for the former 14 classes. Then, based on these three subsets, three tasks are created to mimic the continual learning.

The numbers of images of the three subsets are 10297, 6666 and 4540 respectively. Within each task, the number of positives for each class is listed in Figure 2.



Classes # of Tasks images	person	car	dog	chair	cat	bird	bottle	aero-plane	train	tvmo-nitor	horse	bicycle	pott-edplant	moto-rbike	sofa	dining table	boat	bus	cow	sheep
Task 1	4037	1906	1910	872	1591	1336	738	0	0	0	0	0	0	0	0	0	0	0	0	0
Task 2	2571	381	87	352	89	12	143	1101	1058	907	1031	996	751	965	0	0	0	0	0	0
Task 3	1494	308	128	757	59	29	281	11	6	153	12	38	245	28	959	928	861	781	571	518

Figure 2: Number of images for each class in 3 tasks

For each task, we randomly sample 10% of images as testing data. The number of positive samples for each class in the whole testing set is shown in Figure 1.

The numbers of positive samples for each class in the training set of task 1, 2 and 3 are shown in Figure 3.

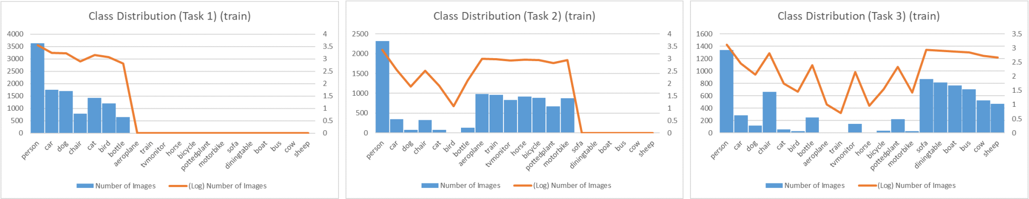


Figure 3: Class distribution in 3 tasks

The number of classes in each image is calculated. The histograms of the number of classes are shown in Appendix Figure 6.

4 Experiments

4.1 Experiment Setting

We use the resnet-18 [4] as our backbone. We extract the hidden layer of pretrained Resnet-18 as our initilized shared parameters. For each tasks, we have task-specific fully-connected layers as task-specific parameters. Our continual learning process starts with an initial learning rate of 10^{-1} and decay with factor 0.1 at the 9^{th} epoch. We train 15 epochs in total based on our model for each individual task to yield our phase-based results. The fisrt task starts at the first epoch, the second starts at the 16^{th} epoch and the third task starts at the 31^{th} epoch.

4.2 Evaluation Configuration and Metrics

We compare our PPI method to some alternative algorithms:

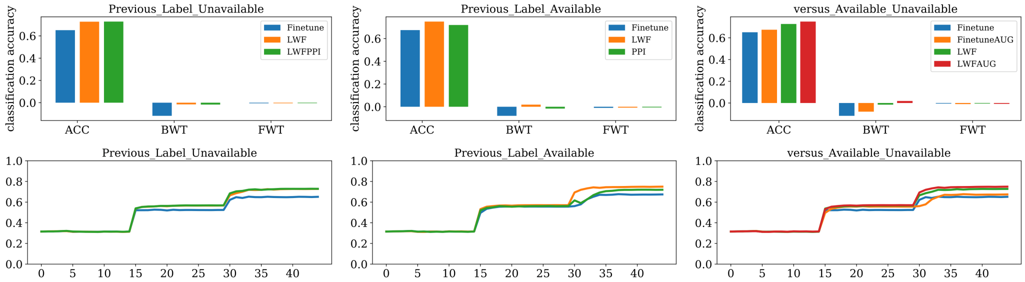


Figure 4: Results in Dataset Variation setting

- Joint Training: we use the model to train the whole dataset and yield the results.
- Finetuning method: we use the finetuning method as our baseline method, which view the three tasks as three independent tasks by using the subset of each task to finetune the model without adding any additional bells-and-whistles.
- LWF: We use the Learning Without Forgetting (LWF) method as one of the technique to be compared. In this paper, we extend the LWF method to handle the multi-lable classification problem.
- LWF+PPI: We integrate the LWF method with our PPI algorithm as the final method.

Futher, to quantify the relative contribution of each setting, the performances of the methods are examined using the following configurations.

- Dataset Variation (DV): during the training process of the current task, we could choose to provide or to not provide the labels related to the former tasks to task-specific input.

4.3 Metrics

For each method, we use the mAP (mean Average Precision) metric to evaluate our result of task 3. Besides monitoring its performance across tasks, it is also important to assess the ability of the learner to transfer knowledge [10].

1. Backward transfer (BWT), which is the influence that learning a task t has on the performance on a previous task $k < t$. On the one hand, there exists positive backward transfer when learning about some task t increases the performance on some preceding task k. On the other hand, there exists negative backward transfer when learning about some task t decreases the performance on some preceding task k. Large negative backward transfer is also known as (catastrophic) forgetting.

2. Forward transfer (FWT),which is the influence that learning a task t has on the performance on a future task $k > t$. In particular, positive forward transfer is possible when the model is able to perform "zero-shot" learning, perhaps by exploiting the structure available in the task descriptors.

4.4 Quantitative Results

The comparison among our methods against other methods are shown in Figure 4 and Figure 5. By considering the Dataset Variation (DV), we could draw that LWFPPI preforms best. When providing the previous positive labels of previous tasks, the LWF performs best. While comparing the results of whether or not give the previous labels, the Finetuning and LWF methods both improve the performance when getting access to the extra data. In Figure 5, by comparing all the methods we use, we get to know the LWFPPI behaves best and during the training process, the performance of classes of task 1 and task 2 may fluctuate for some other methods. The LWFPPI is much robust compared to other all methods when handling the incremental multi-label classification problems

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.

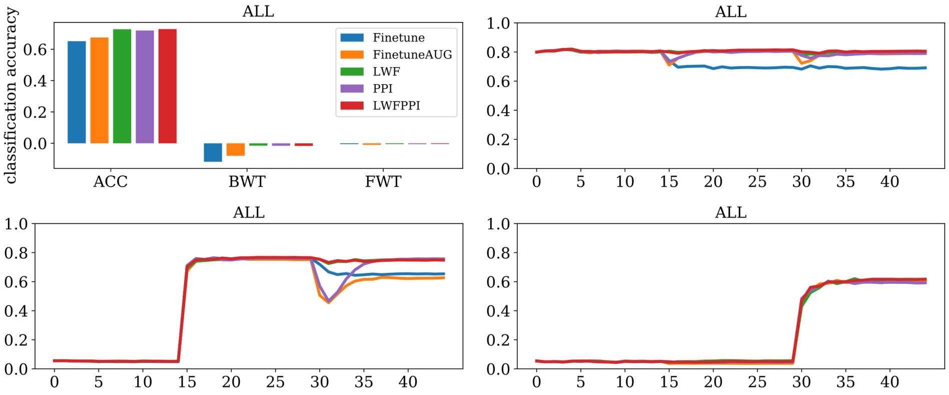


Figure 5: Results of each task for All methods in Data Variation setting

[2] Z. Chen, V. Badrinarayanan, C.-Y. Lee, and A. Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. *arXiv preprint arXiv:1711.02257*, 2017.

[3] 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.

[4] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[5] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin. Lifelong learning via progressive distillation and retrospection. In *European Conference on Computer Vision*, pages 452–467. Springer, 2018.

[6] A. Kendall, Y. Gal, and R. Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics.

[7] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, page 201611835, 2017.

[8] 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, 2017.

[9] Z. Li and D. Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018.

[10] D. Lopez-Paz et al. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017.

[11] M. McCloskey and N. J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109 – 165. Academic Press, 1989.

[12] C. V. Nguyen, Y. Li, T. D. Bui, and R. E. Turner. Variational continual learning. *arXiv preprint arXiv:1710.10628*, 2017.

[13] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. icarl: Incremental classifier and representation learning. In *Proc. CVPR*, 2017.

[14] O. Sener and V. Koltun. Multi-task learning as multi-objective optimization. In *Advances in Neural Information Processing Systems*, pages 524–535, 2018.

[15] J. Serra, D. Suris, M. Miron, and A. Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In J. Dy and A. Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 4548–4557, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[16] H. Shin, J. K. Lee, J. Kim, and J. Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, pages 2990–2999, 2017.

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

A Future Directions

A.1 Inherent relation between two streams of methods

We are interested to see if there are inherent relations between retrospection methods and structural regularization approaches. For example, checking if there are some similarities of optimization trajectories between memory mechanisms and structural regularization. Beside, we can also check how far the important weights deviates from the optimal at the end of previous tasks using retrospection mechanisms.

A.2 Visualization

A.3 Semi-supervised Continual Learning

A.4 Active Continual Learning

A.5 Larger dataset

Try to perform continual learning on Open Image V4 and/or Tencent ML-Images.

A.6 Theoretical guarantees

A.7 Regret analysis

B Misc of dataset

We acknowledge Liping Tang's work on data preprocessing. .

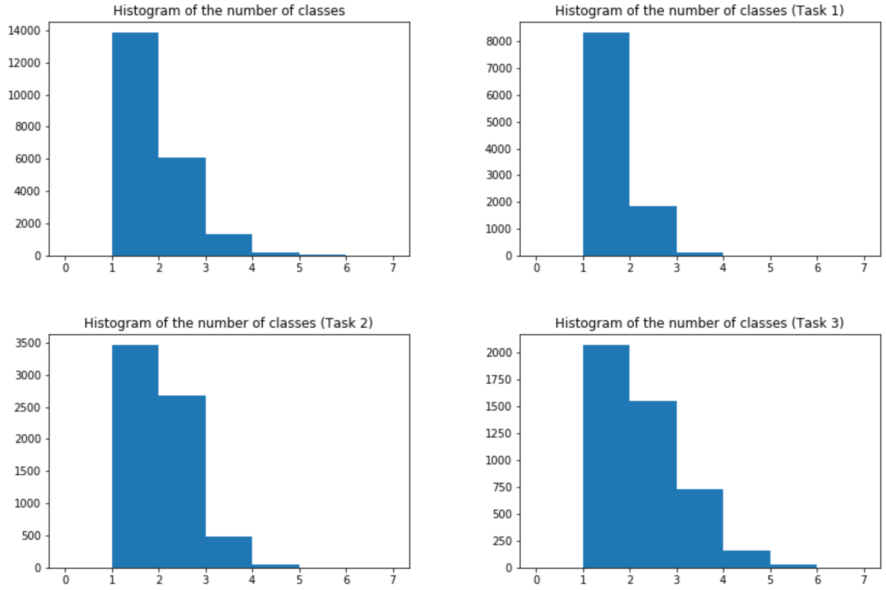


Figure 6: Histograms of the number of classes in each tasks