Compacting, Picking and Growing for Unforgetting Continual Learning

Steven C. Y. Hung, Cheng-Hao Tu, Cheng-En Wu, Chien-Hung Chen, Yi-Ming Chan, and Chu-Song Chen

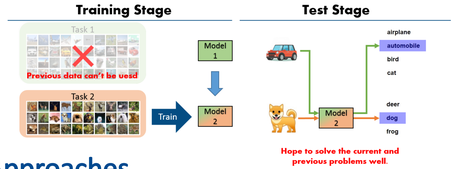
Institute of Information Science, Academia Sinica,
MOST Joint Research Center for AI Technology and All Vista Healthcare (Taipei, Taiwan)

NeurIPS 2019

Introduction

•Continual lifelong learning

Setting: training data of old tasks are non-available for the new tasks. Assume clear task boundaries (i.e., labels non-overlapping).



•Existing Approaches

Regularization (eg. EWC): cannot ensure un-forgetting.

Memory or GAN replay: cannot guarantee the exact performance; replay needs re-training which requires memory.

Dynamic architecture: model is monotonically increased; a redundant structure is yielded.

•Motivation of our approach

Deep learning: a process of turning data to weights.

Model compression: pruning the redundant weights does not affect the neural-net performance.

Compression-selection-expansion loop: We leverage model compression for continual learning. The old-task weights are compressed and remain fixed, but can be picked (via a learnable mask) and trained together with the additional weights released for the new task.

• Characteristics of our method

Avoid forgetting: The function mappings previously built via the compressed models are maintained as exactly the same when new tasks are incrementally added.

Expand with shrinking: Allows model expansion but keeps the compactness of the model; can potentially handle unlimited sequential tasks.

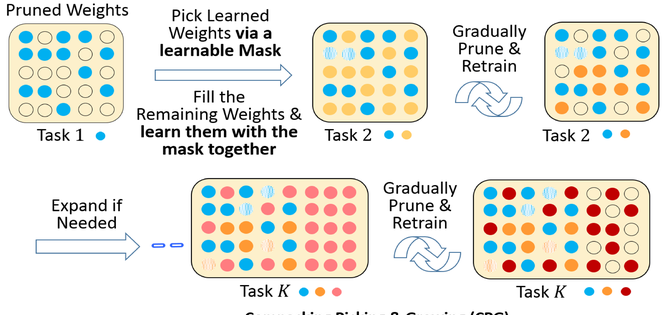
Compact knowledge base: The condensed model recorded for previous tasks serves as knowledge base with accumulated experience for weights picking, yielding performance enhancement for learning new tasks.

Compacting, Picking & Growing (CPG)

•Summary of our method

Our method is designed by combining the ideas of deep model compression via weights pruning (**Compacting**), critical weights selection (**Picking**), and Progressive Net extension (**Growing**).

•Illustration of our approach



Compacking Picking & Growing (CPG)

(√ Avoid forgetting; √ Compactness; √ Extensible; √ Exploiting previous knowledge better)

Algorithm 1: Compacting, Picking and Growing Continual Learning

Input: given task 1 and an original model trained on task 1.
Set an accuracy goal for task 1;
Alternatively remove small weights and re-train the remaining weights for task 1 via gradual pruning [51], whenever the accuracy goal is still hold;
Let the model weights preserved for task 1 be $\mathbf{W}_1^P$ (referred to as task-1 weights), and those that are removed by the iterative pruning be $\mathbf{W}_1^E$ (referred to as the released weights);
for *task* $k = 2 \cdots K$ *(let the released weights of task* k *be* W_k^E*)* **do**
  Set an accuracy goal for task k;
  Apply a mask $\mathbf{M}$ to the weights $\mathbf{W}_{1:k-1}^P$; train both $\mathbf{M}$ and $\mathbf{W}_{k-1}^E$ for task k, with $\mathbf{W}_{1:k-1}^P$ fixed;
  If the accuracy goal is not achieved, expand the number of filters (weights) in the model, reset $\mathbf{W}_{k-1}^E$ and go to previous step;
  Gradually prune $\mathbf{W}_{k-1}^E$ to obtain $\mathbf{W}_k^E$ (with $\mathbf{W}_{1:k-1}^P$ fixed) for task k, until meeting the accuracy goal;
  $\mathbf{W}_k^P = \mathbf{W}_{k-1}^E \backslash \mathbf{W}_k^E$ and $\mathbf{W}_{1:k}^P = \mathbf{W}_{1:k-1}^P \cup \mathbf{W}_k^P$;
end

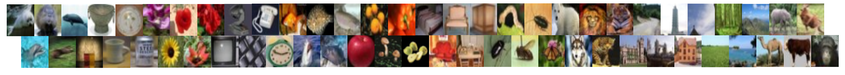
References

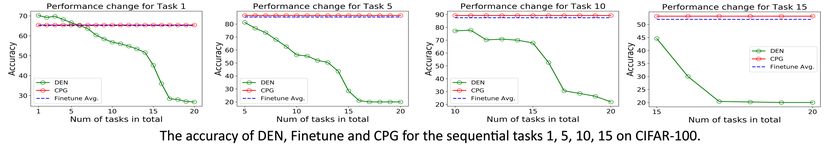
ProgressiveNet [Andrei A Rusu et al. , *arXiv16*], **PackNet** [Arun Mallya et al. , *CVPR18*], **Pack & Expand (PAE)** [Steven CY Hung et al. , *ICMR19*], **Piggyback** [Arun Mallya et al. , *ECCV18*], **Gradual pruning** [Michael Zhu et al. , *ICLR Workshop18*], **DEN** [Jaehong Yoon et al. , *ICLR18*]

Experiments

•20 tasks on CIFAR100 dataset

Divide CIFAR-100 into 20 tasks. Each has 5 classes. (VGG16-BN model)





The accuracy of DEN, Finetune and CPG for the sequential tasks 1, 5, 10, 15 on CIFAR-100.

Methods	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Avg.	Exp. (×)	Red. (×)
PackNet	66.4	80.0	76.2	78.4	80.0	79.8	67.8	61.4	68.8	77.2	79.0	59.4	66.4	57.2	36.0	54.2	51.6	58.8	67.8	83.2	67.5	1	0
PAE	67.2	77.0	78.6	76.0	84.4	81.2	77.6	80.0	80.4	87.8	85.4	77.8	79.4	79.6	51.2	68.4	68.6	68.6	83.2	88.8	77.1	2	0
CPG	65.2	76.6	79.8	81.4	86.6	84.8	83.4	85.0	87.2	89.2	90.8	82.4	85.6	85.2	53.2	74.4	70.0	73.4	88.8	94.8	80.9	1.5	0.41

The performance of PackNet, PAE and CPG on CIFAR-100 twenty tasks.

Methods	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Avg.	Exp. (×)	Red. (×)
Scratch	65.8	78.4	76.6	82.4	82.2	84.6	78.6	84.8	83.4	89.4	87.8	80.2	84.4	80.2	52.0	69.4	66.4	70.0	87.2	91.2	78.8	20	0
fine-Avg	65.2	76.1	76.1	77.8	85.4	82.5	79.4	82.4	82.0	87.4	87.4	81.5	84.6	80.8	52.0	72.1	68.1	71.9	88.1	91.5	78.6	20	0
fine-Max	65.8	76.8	78.6	80.0	86.2	84.8	80.4	84.0	83.8	88.4	89.4	83.8	87.2	82.8	53.6	74.6	68.8	74.4	89.2	92.2	80.2	20	0
CPG avg	65.2	76.6	79.8	81.4	86.6	84.8	83.4	85.0	87.2	89.2	90.8	82.4	85.6	85.2	53.2	74.4	70.0	73.4	88.8	94.8	80.9	1.5	0.41
CPG max	67.0	79.2	77.2	82.0	86.8	87.2	82.0	85.6	86.4	89.6	90.0	84.0	87.2	84.8	55.4	73.8	72.0	71.6	89.6	92.8	81.2	1.5	0
CPG top	66.6	77.2	78.6	83.2	88.2	85.8	82.4	85.4	87.6	90.8	91.0	84.6	89.2	83.0	56.2	75.4	71.0	73.8	90.6	93.6	81.7	1.5	0

The performance of CPGs and individual models on CIFAR-100 twenty tasks.

Exp. : *expansion of weights.*

Red. : *redundant weights.*

Scratch: *each task independently trained from scratch.*

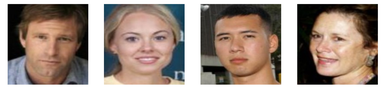
fine-Avg/Max: *average/maximum accuracy of fine-tuning from a previous model randomly selected and repeats the process 5 times.*

CPG avg/max: *setting the accuracy goals to be fine-Avg and fine-Max in CPG, respectively.*

CPG top: *setting the goal to be slightly larger than the maximum of fine-Avg/Max in CPG.*

•Facial-informatic Tasks

Face ➡ Gender ➡ Expression ➡ Age



Task	Train from Scratch	Finetune	CPG
Face	99.417 ± 0.367	-	99.300 ± 0.384
Gender	83.70	90.80	89.66
Expression	57.64	62.54	63.57
Age	46.14	57.27	57.66
Exp. (×)	4	4	1
Red. (×)	0	0	0.003

Accuracy on facial-informatic dataset. (Model: CNN-20)

•Fine-grained Image Tasks

ImageNet ➡ Birds ➡ Cars ➡ Flowers ➡ Paintings ➡ Sketchs

Dataset	Train from Scratch	Finetune	Prog.Net	PackNet	Piggyback	CPG
ImageNet	76.16	-	76.16	75.71	76.16	75.81
CUBS	40.96	82.83	78.94	80.41	81.59	83.59
Stanford Cars	61.56	91.83	89.21	86.11	89.62	92.80
Flowers	59.73	96.56	93.41	93.04	94.77	96.62
Wikiart	56.50	75.60	74.94	69.40	71.33	77.15
Sketch	75.40	80.78	76.35	76.17	79.91	80.33
Model Size (MB)	554	554	563	115	121	121

Accuracy on fine-grained tasks. (Model: ResNet-50)

Code: https://github.com/ivclab/CPG