Learning to Prompt for Continual Learning

1、 Motivation

作者认为当前增量学习面临两个问题：

(1) Whether the form of episodic memory can go beyond buffering past data to more intelligent and succinct episodic memory system?

(2) How to automatically select relevant knowledge component for arbitrary sample without knowing its task identity?

对第一个问题，作者提出使用NLP的最新训练范式prompt-based learning来解决。A prompt encodes task-specific knowledge and has the ability to utilize pre- trained frozen models more effectively than ordinary finetuning. Thus, it is promising to leverage prompts to learn knowledge, and further store learned knowledge, in the continual learning context.

对第二个问题，如果对不同的任务训练不同的prompts，那么事实上还是需要知道task identity。但如果只训练一个prompt，那么可能还是会有灾难性遗忘的问题。As a transfer learning technique, the target of prompting is to make frozen pre-trained models achieve good performance on down-streaming individually, not sequentially. There- fore, if we instead maintain a single shared prompt for all tasks, the problem of catastrophic forgetting may still exist.

为了解决这一问题，作者提出了新的增量学习方法：

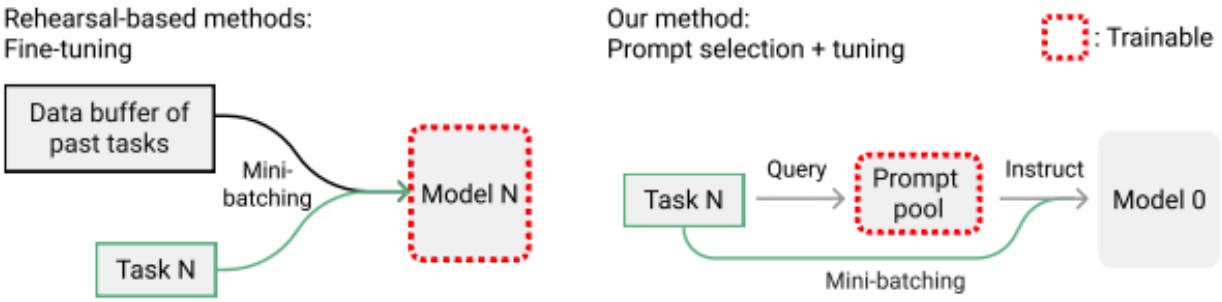

Figure 1. Overview of the L2P framework. Compared with typical methods that adapt entire or partial model weights to tasks sequentially with a rehearsal buffer to avoid forgetting, L2P uses a single backbone model and learns a prompt pool to instruct the model conditionally. Task-specific knowledge is stored inside a prompt pool, thus a rehearsal buffer is no longer mandatory to mitigate forgetting. L2P automatically selects and updates prompts from the pool in an instance-wise fashion, thus task identity is not required at test time. Notably, our largest prompt space is smaller than the size of one 224×224 image.

instead of tuning the parameters during the continual learning process, L2P **keeps the pre-trained model untouched**, and instead **learns a set of prompts that dynamically instruct models to solve corresponding tasks**. 意思是要训练一系列prompts，而不用训练model

2、Approach

2.1 From prompt to prompt pool

我认为下面这段描述能比较好地描述作者需要解决的问题：

First, the task identity at test time is unknown so training task-independent prompts is not feasible.

Second, even if the task-independent prompt can be known at test time, it prevents possible knowledge sharing between simi- lar tasks.

Third, while the naive way of learning a sin- gle shared prompt for all tasks enables knowledge sharing, it still causes severe forgetting issue

Ideally one would learn a model that is able to share knowledge when tasks are similar, while maintaining knowledge independent otherwise.

基于这样的想法，作者设计了**prompt pool** $P = \{P_1, \dots, P_M\}$。其中，每个独立的prompt表示为 $P_j \in \mathbb{R}^{L_p \times D}$。对于一个输入x，需要自适应地选出N个与之task相关的prompts，假设选到的prompy序列为 $\{s_i\}_{i=1}^{N}$那么输入预训练好的Transformer Encoder的输入可以表示为$x_p = [P_{s1}; \dots; P_{sN}; x_e], 1 \le N \le M$。

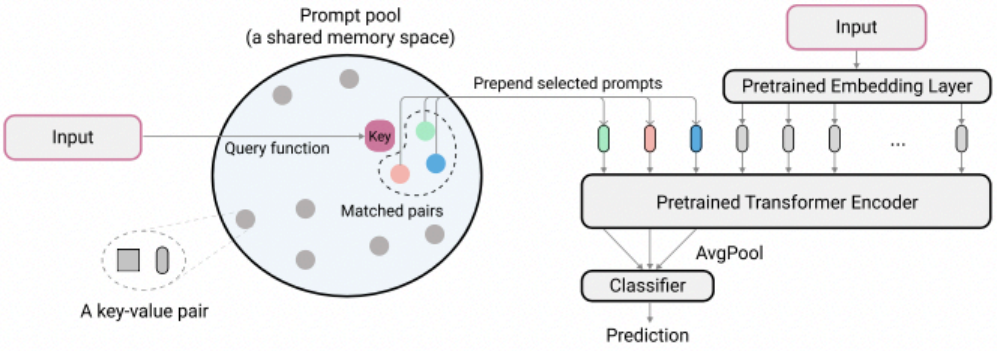

Figure 2. Illustration of L2P at test time. We follow the same procedure at training time: First, L2P selects a subset of prompts from a key-value paired *prompt pool* based on our proposed instance-wise query mechanism. Then, L2P prepends the selected prompts to the input tokens. Finally, L2P feeds the extended tokens to the model, and optimize the prompt pool through the loss defined in equation 5. The objective is learning to select and update prompts to instruct the prediction of the pre-trained backbone model.

2.2 Instance-wise prompt query

一个输入需要对应N个prompt，那么怎么选出这些prompt呢？为此作者设计了一个**基于key-value对的查询策略**。

将每一个prompt与一个key关联：$\{(k_1, P_1), \dots, (k_M, P_M)\}$，其中$k_M \in \mathbb{R}^{D_k}$，所有key的集合用K表示。

对于一个输入x，用一个query function $q : \mathbb{R}^{H \times W \times C} \to \mathbb{R}^{D_k}$将其映射到与key相同的维度。要注意的是这里的q应该是一个固定的不可学习的函数。因此用了一个已经训练好的模型来充当q。

接着，用计算query与任意N个keys之间的余弦距离之和$\gamma()$，选距离最小的作为被选中的keys。

$$K_x = argmin_{\{s_i\}_{i=1}^{N} \subseteq [1,M]} \sum_{i=1}^{N} \gamma(q(x), k_{s_i})$$

Optionally diversifying prompt-selection.

此外，无论在实验场景还是真实场景，训练的过程中知道task边界是很正常的，因此作者也考虑加入这个先验。

在训练的过程中，维护一个prompt激活表$H_t = [h_1, \ldots, h_M]$，其中每个值表示直到第t-1个task，对应prompt激活的**归一化频率**。为了让查询机制尽可能选到不同的prompt，将上面的式子调整为：

$$K_x = argmin_{\{s_i\}_{i=1}^N \subseteq [1,M]} \sum_{i=1}^{N} \gamma(q(x), k_{s_i}) \cdot h_{s_i}$$

这个式子仅在训练的时候使用，在测试的时候不使用。

为什么测试的时候不用呢?

2.3 optimization

最终损失函数可以表示为:

$$\min_{\mathbf{P},\mathbf{K},\phi} \quad \mathcal{L}(g_\phi(f_r^{\text{avg}}(\boldsymbol{x}_p)), y) + \lambda \sum_{\mathbf{K}_{\boldsymbol{x}}} \gamma\left(q(\boldsymbol{x}), \boldsymbol{k}_{s_i}\right),$$
$$s.t., \quad \mathbf{K}_{\boldsymbol{x}} \text{ is obtained with equation 3},$$

The first term is the softmax cross-entropy loss, the second term is a surrogate loss to pull selected keys closer to corresponding query features. λ is a scalar to weight the loss.

3、Experiment

Results on class-incremental learning.

Table 1. Results on class-incremental learning (i.e., task identity is unknown at test time). Compared methods are grouped based on different rehearsal buffer sizes. 0 means no rehearsal is required, where most SOTA methods are not applicable anymore. Importantly, L2P can attain competitive results without it and greatly outperform them with a small buffer size.

Method	Buffer size	Split CIFAR-100		Buffer size	5-datasets	
		Average Acc (↑)	Forgetting (↓)		Average Acc (↑)	Forgetting (↓)
FT-seq-frozen	0	17.72±0.34	59.09±0.25	0	39.49±0.12	42.62±0.20
FT-seq		33.61±0.85	86.87±0.20		20.12±0.42	94.63±0.68
EWC [21]		47.01±0.29	33.27±1.17		50.93±0.09	34.94±0.07
LwF [28]		60.69±0.63	27.77±2.17		47.91±0.33	38.01±0.28
L2P (ours)		**83.83±0.04**	**7.63±0.30**		**81.14 ±0.93**	**4.64 ±0.52**
ER [8]	10/class	67.87±0.57	33.33±1.28	5/class	80.32±0.55	15.69±0.89
GDumb [46]		67.14±0.37	-		56.99±0.06	-
BiC [61]		66.11±1.76	35.24±1.64		78.74±1.41	21.15±1.00
DER++ [3]		61.06±0.87	39.87±0.99		80.81±0.07	14.38±0.35
Co^2L [4]		72.15±1.32	28.55±1.56		82.25±1.17	17.52±1.35
L2P-R (ours)		**84.21±0.53**	**7.72±0.77**		**85.56±0.95**	**4.22±0.03**
ER [8]	50/class	82.53±0.17	16.46±0.25	10/class	84.26±0.84	12.85±0.62
GDumb [46]		81.67±0.02	-		70.76±0.12	-
BiC [61]		81.42±0.85	17.31±1.02		85.53±2.06	10.27±1.32
DER++ [3]		83.94±0.34	14.55±0.73		84.88±0.57	10.46±1.02
Co^2L [4]		82.49±0.89	17.48±1.80		86.05±1.03	12.28±1.44
L2P-R (ours)		**86.31±0.59**	**5.83±0.61**		**88.95±0.78**	**4.92±0.71**
Upper-bound	-	90.85±0.12	-	-	93.93±0.18	-

buffer size越小，提升越明显。

Table 2. Class-incremental results on Split CIFAR-100 against architecture-based methods. `Diff = Upper-Bound Acc - Method Acc` (lower is better) measures how close the performance to the upper-bound of the used backbone.

Method	**Backbone**	**Avg. Acc (↑)**	**Diff (↓)**
Upper-bound	ResNet18	80.41	-
SupSup [60]		28.34±2.45	52.07
DualNet [44]		40.14±1.64	40.27
Upper-bound	ViT-B/16	90.85	-
L2P (ours)		**83.83**±0.04	**7.02**

Results on domain-incremental learning

Table 3. Results on domain-incremental learning using CORe50 dataset, in terms of test accuracy.

Method	**Buffer size**	Test Acc (↑)
EWC [21]	0	74.82±0.60
LwF [28]		75.45±0.40
L2P (ours)		**78.33**±0.06
ER [8]	50/class	80.10±0.56
GDumb [46]		74.92±0.25
BiC [61]		79.28±0.30
DER++ [3]		79.70±0.44
Co^2L [4]		79.75±0.84
L2P-R (ours)		**81.07**±0.13
Upper-bound	-	82.15±0.37

Results on task-agnostic learning.

Table 4. Results on task-agnostic continual learning using the Gaussian scheduled CIFAR-100 dataset, in terms of test accuracy.

Method	**Buffer size**	Test Acc (↑)
EWC [21]	0	63.04±0.42
L2P (ours)		**88.34±0.14**
ER [8]	50/class	82.63±0.27
GDumb [46]		81.67±0.02
DER++ [3]		85.24±0.71
L2P-R (ours)		**88.92±0.39**
Upper-bound	-	90.85±0.12

Ablation study

Table 5. Ablation studies. See text for detailed explanations.

Ablated components	**5-datasets**	
	Average Acc (↑)	Forgetting (↓)
w/o prompt pool	51.96	26.60
w/o key-value pair	58.33	20.45
w/o diversified selection	62.26	17.84
none	**81.14**	**4.64**

第一行是只用一个prompt去训练的结果；

第二行是将N个prompts的均值作为key，而不是定义一个key。

第三行就是把dibersified selection去掉。

visualization

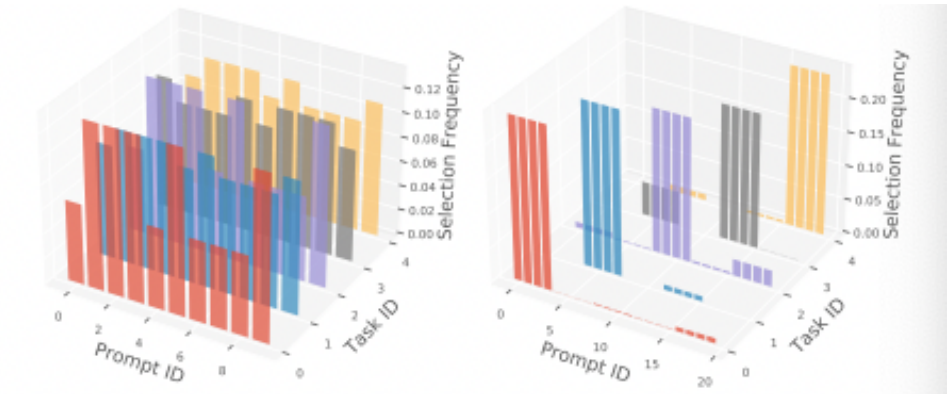

Figure 3. Prompt selection histograms for (left) Split CIFAR-100 and (right) 5-datasets. CIFAR-100 has higher intra-task similarity compared to 5-datasets, thus largely sharing prompts between tasks results in good performance, while 5-datasets favors more task-specific prompts. We only show the first 5 tasks for Split CIFAR-100 for better readability.