Memory Aware Synapses: Learning what (not) to forget

1、Motivation

如果一个人能够在学习的过程中反复地学习以前学过的东西，那么他会记得更牢。这个过程其实是“将有用的东西记住，没用的东西遗忘”的过程。基于这个直觉，作者认为可以在每一个task到来之前进行一个deployment的操作，使用无标签的数据让模型学会过往的知识那些对他是重要的，从而强化这部分记忆，遗忘不重要的部分。

2、Approach

2.1 Estimating parameter importance

对于已经训练好的一个对于输入X可以得到输出Y的函数$\bar{F}$的估计F，要度量这个函数对改变其参数的敏感度。用这个敏感度表示该参数的重要性。计算方法如下：

$$F(x_k;\theta+\delta)-F(x_k;\theta)\approx\sum_{i,j}g_{ij}(x_k)\delta_{i,j}$$

$$\Omega_{ij}=\frac{1}{N}\sum_{k=1}^{N}||g_{ij}(x_k)||$$

其中$g_{ij}(x_k)=\frac{\partial(F(x_k;\theta))}{\partial\theta_{ij}}$。上一个式子是梯度的表示，下面式子是某一参数对所有输入梯度的累加。如果Ω_{ij}比较小，说明其对应的参数对输出的影响不大。反之，如果其比较大，说明在后面的训练中需要将对应的参数保护起来。

2.2 Learning a new task

当第T_n个task到来的时候，loss的计算表示为：

$$L(\theta)=L_n(\theta)+\lambda\sum_{i,j}\Omega_{ij}(\theta_{ij}-\theta^*_{ij})^2$$

在完成新task的训练后，再更新Ω矩阵。这个更新的过程可以在训练完成后的任何时候，因此更为灵活。且不需要label作为监督。实验中超参数λ设置为1。

2.3 Connection to hebbian learning

A local version of our method.

作者将网络看成一系列函数的组合：$F(x)=F_L(F_{L-1}(\dots(F_1(x))))$。

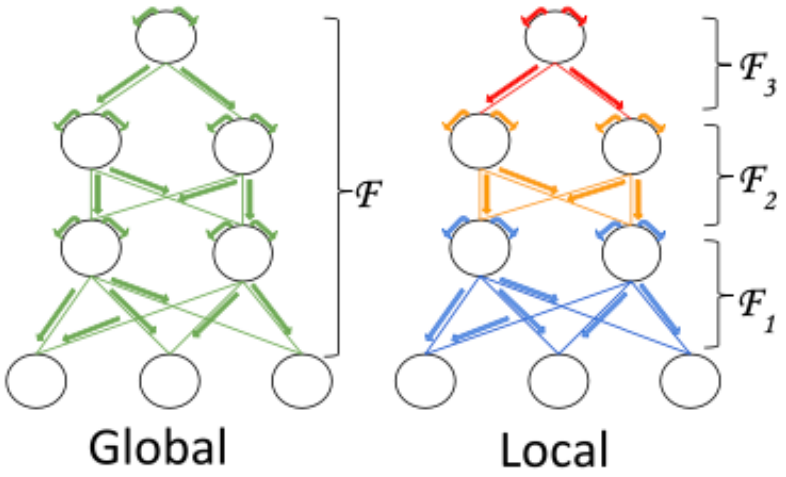


Fig. 3: Gradients flow for computing the importance weight. Local considers the gradients of each layer independently.

这样设计就可以局部地保护参数，但是这样设计相比global有什么好处呢?

3、Experiment

two task experiment

Method	Birds → Scenes		Scenes → Birds		Flower → Birds		Flower → Scenes	
FineTune	45.20 (-8.0)	**57.8**	49.7 (-9.3)	**52.8**	64.87 (-13.2)	**53.8**	70.17 (-7.9)	57.31
LwF [17]	51.65 (-2.0)	55.59	55.89 (-3.1)	49.46	73.97 (-4.1)	53.64	76.20 (-1.9)	58.05
EBLL [28]	52.79 (-0.8)	55.67	56.34 (-2.7)	49.41	75.45 (-2.6)	50.51	76.20 (-1.9)	**58.35**
IMM [16]	51.51 (-2.1)	52.62	54.76 (-4.2)	52.20	75.68 (-2.4)	48.32	76.28 (-1.8)	55.64
EWC [12]	52.19 (-1.4)	55.74	**58.28 (-0.8)**	49.65	76.46 (-1.6)	50.7	77.0 (-1.1)	57.53
SI [39]	52.64 (-1.0)	55.89	57.46 (-1.5)	49.70	75.19 (-2.9)	51.20	76.61 (-1.5)	57.53
MAS (ours)	**53.24 (-0.4)**	55.0	57.61 (-1.4)	49.62	**77.33 (-0.7)**	50.39	**77.24 (-0.8)**	57.38

Table 2: Classification accuracy (%), drop in first task (%) for various sequences of 2 tasks using the object recognition setup.

希望的效果是在训练的时候对前面任务对遗忘能够尽可能地小，但在新任务上的准确率又可以尽可能地逼近finetune的效果。

Local vs. global MAS on training/test data.

Method	Ω_{ij} computed. on	Birds → Scenes		Scenes → Birds		Flower → Bird		Flower → Scenes	
MAS	Train	53.24 (-0.4)	55.0	57.61 (-1.4)	49.62	77.33 (-0.7)	50.39	77.24 (-0.8)	57.38
MAS	Test	53.43 (-0.2)	55.07	57.31 (-1.7)	49.01	77.62 (-0.5)	50.29	77.45 (-0.6)	57.45
MAS	Train + Test	53.29 (-0.3)	56.04	57.83 (-1.2)	49.56	77.52 (-0.6)	49.70	77.54 (-0.5)	57.39
l-MAS	Train	51.36 (-2.3)	55.67	57.61 (-1.4)	49.86	73.96 (-4.1)	50.5	76.20 (-1.9)	56.68
l-MAS	Test	51.62 (-2.0)	53.95	55.74 (-3.3)	50.43	74.48 (-3.6)	50.32	76.56 (-1.5)	57.83
l-MAS	Train + Test	52.15 (-1.5)	54.40	56.79 (-2.2)	48.92	73.73 (-4.3)	50.5	76.41 (-1.7)	57.91

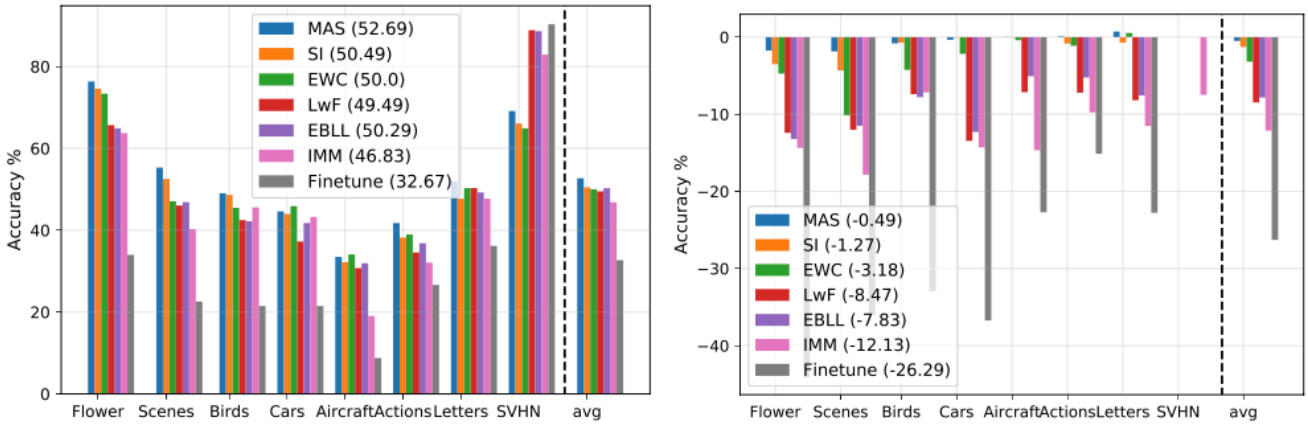
Table 3: Classification accuracies (%) for the object recognition setup - comparison between using Train and Test data (unlabeled) to compute the parameter importance Ω_{ij}.

Local version虽然在计算开销上有优势，但是带来的问题是准确率比不上global version。

l2 vs. vector output.

使用l2速度更快，效果上没有明显的下降。

longer sequence



(a) (b)

Fig. 5: 5a performance on each task, in accuracy, at the end of 8 tasks object recognition sequence. 5b drop in each task relative to the performance achieved after training each task.

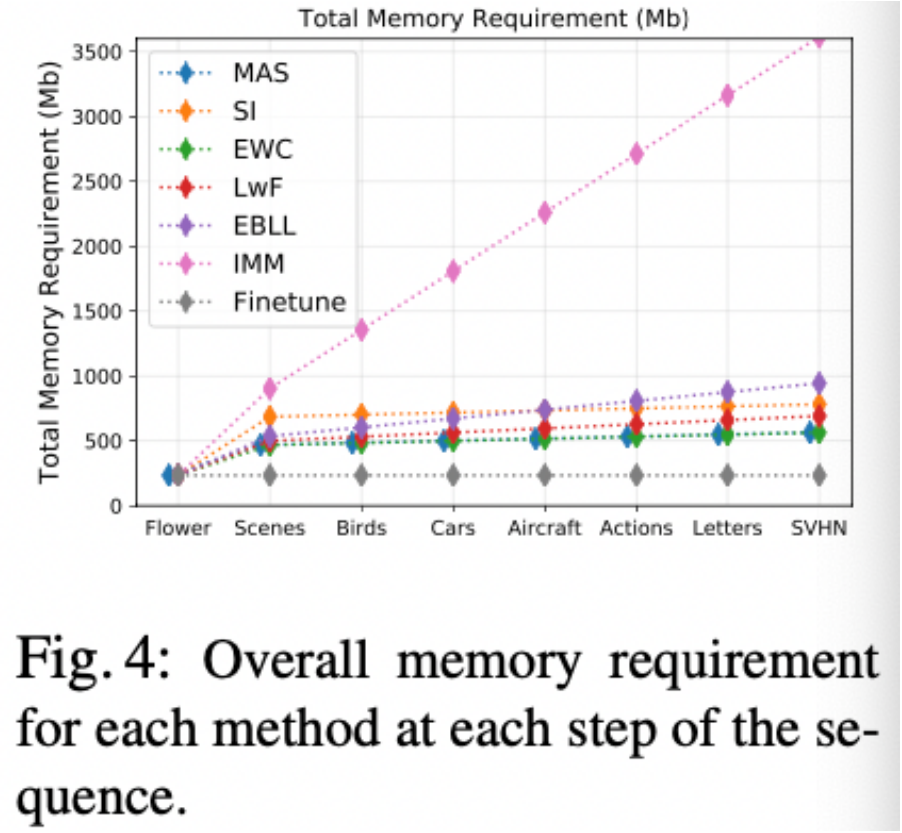


Fig. 4: Overall memory requirement for each method at each step of the sequence.

文章的方法不仅在长序列上效果较好而且内存开销也最接近finetune。