Transfer without forgetting (2023)

November 7, 2023 10:30 AM

★ **Main idea:**

They uses a fixed pretrained model, and they propagate the knowledge to learn new tasks using a layer-wise loss term.

★ **Main findings:**

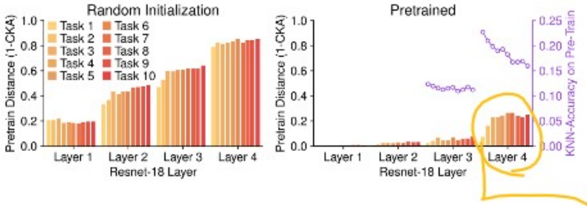


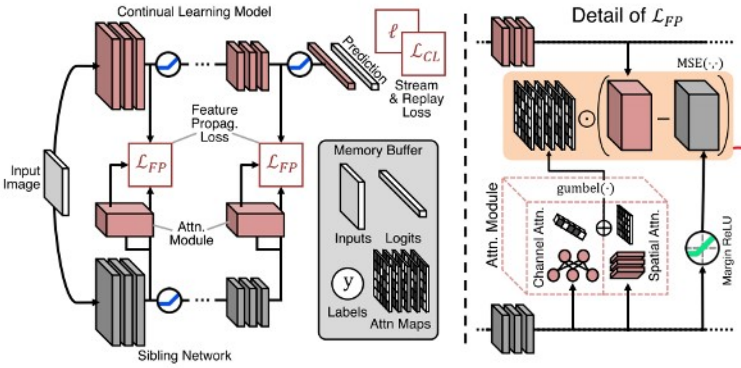
Fig. 1. Forgetting of the initialization, measured as the distance from the pretrain (1−CKA [32]) (lower is better) and kNN accuracy (higher is better). Features extracted by a pretrained model remain closer to the initialization w.r.t. a randomly initialized model. Furthermore, the steady decrease in kNN accuracy as training progresses reveals that features become less specific for past tasks.

Pretraining incurs catastrophic forgetting:

In this figure, they were experimenting with ResNet18 and split Cifar 100 (10 tasks). They wanted to measure how each layer differs from its initialization, whether this initialization was random or pretrained. (lower difference means less forgetting thus that's better). *[but they don't keep the network fixed]*

To sum up, while **pretraining is certainly beneficial,** the model drifts away from it **one task** after the other. Hence, ***only the first task takes full advantage*** of it; the optimization of later tasks, instead, starts from an initialization that increasingly differs from the one attained by pretraining. This is detrimental, as classes introduced later might be likewise advantaged by the reuse of different pieces of the initial knowledge.
[look at layer 4, pretrain distance for task 1 is small (thus less forgetting), but it gets higher with more introduced tasks. Here, they are starting with a fixed pretrained model and then finetune it on subsequent tasks. For this, only initial tasks are getting the benefits but then the forgetting is increased.]

Network architecture:



1. The use a pretrained network and take a copy of it; call it sibling network. This sibling network is always fixed but the other base network is always updated with every new task.
2. Considering a subset of L layers, they seek to minimize the distance between the activations of the base network and those from its (RelUm) pretrained sibling network, as this:

$$\mathop{\mathbb{E}}_{x\sim\mathcal{T}_c}\left[\sum_{l=1}^{L}||h_\theta^{(l)} - \mathrm{ReLU}_\mathrm{m}(\widehat{h}^{(l)})||_2^2\right],$$

Base network Sibling network

Note: this is a form of knowledge distillation to maximize the knowledge transfer between two networks.

Fig. 2. Overview of TwF and detail of $\mathcal{L}_{\rm FP}$: Given a batch of samples from the current task or from $\mathcal{B}$, we *i)* extract intermediate features from both the student and fixed sibling backbones at multiple layers; *ii)* compute the corresponding binarized attention maps $\mathbb{M}(\cdot)$; *iii)* pull the attention-masked representations of the two models closer.

3. The authors are worring about this objective, which places too much emphasis on maximizing knowledge transfer from the sibling model, and that might lead to a problem of excessive rigidity. In other words, the model may become too fixed or inflexible in its behavior, making it less adaptable to the specific data or requirements of the current task.

4. The objective function is updated as this:

Gate == dot

$$\mathop{\mathbb{E}}_{x\sim\mathcal{T}_c}\left[\sum_{l=1}^{L}||\mathbb{M}(\widehat{h}^{(l)})\odot\left(h_\theta^{(l)} - \mathrm{ReLU}_\mathrm{m}(\widehat{h}^{(l)})\right)||_2^2\right]$$

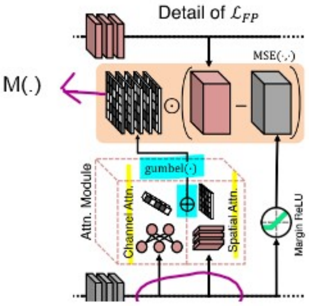
It can be seen that, TwF learns attention map M(·) over the feature maps of the sibling network. M(.) is used as a gating mechanism.

In this context, the gating mechanism helps to make the model more flexible and adaptable by selecting and aligning specific spatial regions of feature maps in a more controlled and fine-grained manner. This selective attention allows the model to focus on the most relevant information and adapt to the specific data and requirements of the current task, preventing it from becoming excessively rigid.

How to learn M(.)?

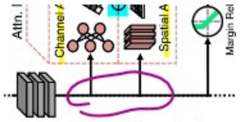
The attention maps M(·) are computed through specific layers, whose architectural design follows some previous works.
Specifically, they:

- forward the input activation maps of the sibling into two parallel branches, producing respectively a Channel Attention MCh(·) map and a Spatial Attention MSp(·) map.
- these two intermediate results are summed and then activated through a binary Gumbel-Softmax sampling, which allows to model discrete on-off decisions regarding which information we want to propagate.



MSp(·) map.

- these two intermediate results are summed and then activated through a binary Gumbel-Softmax sampling, which allows to model discrete on-off decisions regarding which information we want to propagate.



Parallel operations

Total objective function:

$$\min_{\theta,\phi} \underset{(x,y)\sim\mathcal{T}_c}{\mathbb{E}} \left[\ell(y_j^i, f_{(\theta,\phi)}(x_j^i))\right] + \overset{2}{\mathcal{L}_{\mathrm{CL}}} + \overset{3}{\mathcal{L}_{\mathrm{FP}}} + \overset{1}{\mathcal{L}_{\mathrm{AUX}}}.$$

1

$$\mathcal{L}_{\mathrm{AUX}} \triangleq -\lambda \sum_{l=1}^{L} \underset{x_1,\ldots,x_n\sim\mathcal{T}_c}{\mathbb{E}} \left[\sum_{j=1}^{n} \log \frac{e^{g_{ij}^{\mathrm{T}} g_{ij}/T}}{\frac{1}{n}\sum_{k=1}^{n} e^{g_{ij}^{\mathrm{T}} g_{ik}/T}}\right],$$
$$g_{ij} \triangleq \mathrm{NORM}(\mathrm{GAP}(\mathbb{M}(\widehat{h}^{(l)}(x_j)))),$$

The auxiliary loss is used to ensure that the gating maps don't learn simple behaviours (always on or off). This loss assigns different propagation gating masks to different examples. The intuition is that each example has its own preferred subset of channels to be forwarded from the sibling.

2

$$\mathcal{L}_{\mathrm{CL}} \triangleq \underset{(x,y,l)\sim\mathcal{B}}{\mathbb{E}} \left[\alpha \cdot ||f_{(\theta,\phi)}(x) - l||_2^2 + \beta \cdot \ell(y, f_{(\theta,\phi)}(x))\right],$$

This loss is used for knowledge replay (not very important to achieve robust results. They used a small buffer with some samples from previous tasks to avoid forgetting. It ensures current model can still perform well on previous samples.

3

$$\mathcal{L}_{\mathrm{FP}} \triangleq \underset{\substack{(x,t=c)\sim\mathcal{T}_c \\ (x;t)\sim\mathcal{B}}}{\mathbb{E}} \left[\sum_{l=1}^{L} ||\mathbb{M}(\widehat{h}^{(l)};t) \odot \left(h^{(l)} - \mathrm{ReLU_m}(\widehat{h}^{(l)})\right)||_2^2\right]$$
$$+ \underset{\substack{(x,t,m)\sim\mathcal{B} \\ l=1,\ldots,L}}{\mathbb{E}} \left[\mathrm{BCE}\left(\mathbb{M}(\widehat{h}^{(l)};t), m^{(l)}\right)\right],$$

This feature propagated loss has two terms. The first one is what we discussed earlier (maximizing knowledge transfer) whereas the second term is BCE. (m) is the previous attention maps of samples from older tasks. This BCE term helps distill and maintain past attention maps, which contributes to preserving knowledge from previous tasks.

Note that loss 2 is related to CL, where it's only applied on samples from buffer. The other losses are applied on both current and stored samples.

★ Datasets and experiments:

- For performance evaluation, they used average final accuracy and final forgetting.
- For scenarios and datasets, they described the transfer of knowledge from the pretrain by facilitating the amount of similarity between the two distributions.
- *Scenario A: High similarity* - They use **CIFAR-100 as the pretrain dataset** and then **evaluate the models on Split CIFAR-10 (5 binary tasks)**
- Results of this:

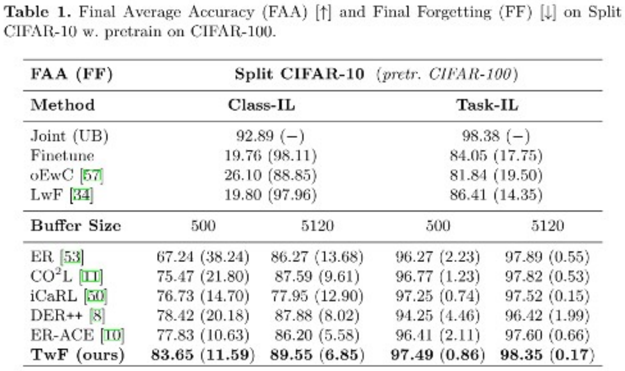


Table 1. Final Average Accuracy (FAA) [↑] and Final Forgetting (FF) [↓] on Split CIFAR-10 w. pretrain on CIFAR-100.

FAA (FF)	**Split CIFAR-10** *(pretr. CIFAR-100)*			
Method	**Class-IL**		**Task-IL**	
Joint (UB)	92.89 (−)		98.38 (−)	
Finetune	19.76 (98.11)		84.05 (17.75)	
oEwC [57]	26.10 (88.85)		81.84 (19.50)	
LwF [34]	19.80 (97.96)		86.41 (14.35)	
Buffer Size	500	5120	500	5120
ER [53]	67.24 (38.24)	86.27 (13.68)	96.27 (2.23)	97.89 (0.55)
CO^{2}L [11]	75.47 (21.80)	87.59 (9.61)	96.77 (1.23)	97.82 (0.53)
iCaRL [50]	76.73 (14.70)	77.95 (12.90)	97.25 (0.74)	97.52 (0.15)
DER++ [8]	78.42 (20.18)	87.88 (8.02)	94.25 (4.46)	96.42 (1.99)
ER-ACE [10]	77.83 (10.63)	86.20 (5.58)	96.41 (2.11)	97.60 (0.66)
TwF (ours)	**83.65 (11.59)**	**89.55 (6.85)**	**97.49 (0.86)**	**98.35 (0.17)**

Note that all competitors undergo an initial pretraining phase prior to CL, thus ensuring a fair comparison.

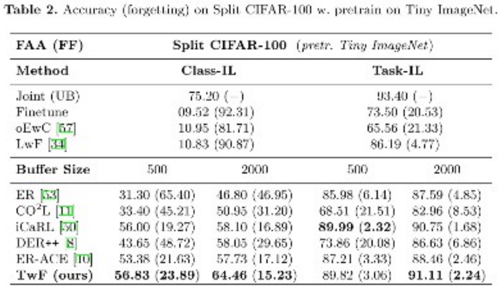


Table 2. Accuracy (forgetting) on Split CIFAR-100 w. pretrain on Tiny ImageNet.

FAA (FF)	**Split CIFAR-100** *(pretr. Tiny ImageNet)*			
Method	**Class-IL**		**Task-IL**	
Joint (UB)	75.20 (−)		93.40 (−)	
Finetune	09.52 (92.31)		73.50 (20.53)	
oEwC [57]	10.95 (81.71)		65.56 (21.33)	
LwF [34]	10.83 (90.87)		86.19 (4.77)	
Buffer Size	500	2000	500	2000
ER [53]	31.30 (65.40)	46.80 (46.95)	85.98 (6.14)	87.59 (4.85)
CO^{2}L [11]	33.40 (45.21)	50.95 (31.20)	68.51 (21.51)	82.96 (8.53)
iCaRL [50]	56.00 (19.27)	58.10 (16.89)	**89.99 (2.32)**	90.75 (1.68)
DER++ [8]	43.65 (48.72)	58.05 (29.65)	73.86 (20.08)	86.63 (6.86)
ER-ACE [10]	53.38 (21.63)	57.73 (17.12)	87.21 (3.33)	88.46 (2.46)
TwF (ours)	**56.83 (23.89)**	**64.46 (15.23)**	89.82 (3.06)	**91.11 (2.24)**

- *Scenario B: Low similarity* - pretrained on tiny-imagenet and evaluated on split cifar 100.

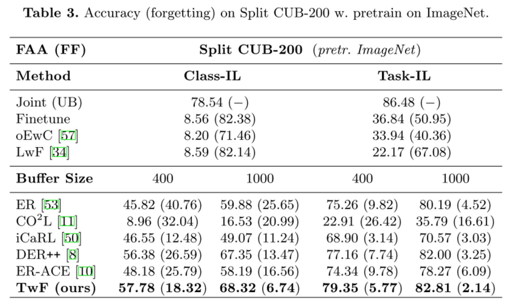


Table 3. Accuracy (forgetting) on Split CUB-200 w. pretrain on ImageNet.

FAA (FF)	**Split CUB-200** (*pretr. ImageNet*)			
Method	**Class-IL**		**Task-IL**	
Joint (UB)	78.54 (−)		86.48 (−)	
Finetune	8.56 (82.38)		36.84 (50.95)	
oEwC [57]	8.20 (71.46)		33.94 (40.36)	
LwF [34]	8.59 (82.14)		22.17 (67.08)	
Buffer Size	400	1000	400	1000
ER [53]	45.82 (40.76)	59.88 (25.65)	75.26 (9.82)	80.19 (4.52)
CO^{2}L [11]	8.96 (32.04)	16.53 (20.99)	22.91 (26.42)	35.79 (16.61)
iCaRL [50]	46.55 (12.48)	49.07 (11.24)	68.90 (3.14)	70.57 (3.03)
DER++ [8]	56.38 (26.59)	67.35 (13.47)	77.16 (7.74)	82.00 (3.25)
ER-ACE [10]	48.18 (25.79)	58.19 (16.56)	74.34 (9.78)	78.27 (6.09)
TwF (ours)	**57.78 (18.32)**	**68.32 (6.74)**	**79.35 (5.77)**	**82.81 (2.14)**

- *Scenario C: Low similarity* - pretrained on ImageNet and evaluated on Split CUB-200.

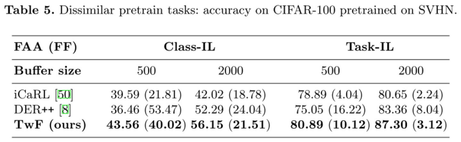


Table 5. Dissimilar pretrain tasks: accuracy on CIFAR-100 pretrained on SVHN.

FAA (FF)	**Class-IL**		**Task-IL**	
Buffer size	500	2000	500	2000
iCaRL [50]	39.59 (21.81)	42.02 (18.78)	78.89 (4.04)	80.65 (2.24)
DER++ [8]	36.46 (53.47)	52.29 (24.04)	75.05 (16.22)	83.36 (8.04)
TwF (ours)	**43.56 (40.02)**	**56.15 (21.51)**	**80.89 (10.12)**	**87.30 (3.12)**

- *Scenario D: Low similarity* - pretrained on SVHN and evaluated on Split Cifar 100.

The paper conducts different and relevant ablation studies. Please refer to them for more details.