Do Pre-trained Models Benefit Equally in Continual Learning?

November 7, 2023 9:08 PM

Main idea:

This paper investigates the effect of pretraining on existing CL methods. introduce a simple yet effective The authors proposed a new simple baseline that employs minimum regularization and leverages the more beneficial pre-trained model, coupled with a two-stage training pipeline.

Main findings:

- It makes more sense to grab a pretrained of-the-shelf network when applying any CL algorithm in real-life.
- It has been proofed (figure to the right) that different CL methods receive different benefits from pretraining. That is, an underperforming algorithm could become competitive and even achieve state- of-the-art performance, when all algorithms start from a pre-trained model.
- Pretraining is especially important when training data is small.

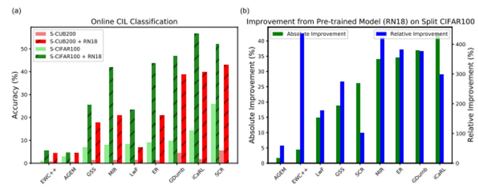

Figure 1. (a) CL algorithms trained from scratch fail on Split CUB200, a more complex dataset than Split CIFAR100, which necessitates the use of pre-trained models (denoted as '+ RN18') that dramatically increase the accuracy of a wide spectrum of algorithms. (b) Different CL algorithms receive vastly different benefits from pre-trained models, and the superiority between algorithms changes. These findings suggest that it is critical for the community to develop CL algorithms with a pre-trained model and understand their behaviors. **[Best viewed in color.]**

Analysis:

Axis	Configurations
Pre-trained Models (7)	Reduced RN18, RN18, RN50, CLIP RN50, SimCLR RN50, SwAV RN50, Barlow Twins RN50
CL Algorithms (11)	ER, MIR, GSS, iCaRL, GDumb, SCR, LwF, EWC++, AGEM, Co^{2}L, DER++
CL Scenarios (2)	CIL, Online CIL

Table 1. We conduct the analyses of pre-trained models in CL by dissecting the space into three axes: 1) different pre-trained models, 2) different CL algorithms, and 3) different CL scenarios.

Total of C classes that are split into T tasks (no overlapping)

Reduced RN18 (R-RN18). ResNet18 whose number of channels is reduced [25] compared with a standard one, which is used in the experiment of training from scratch.
ImageNet Pre-trained RN18, RN50. ResNets pre-trained on ImageNet [15].
CLIP Pre-trained RN50. ResNet50 pre-trained on the WebImageText dataset based on Contrastive Language–Image Pre-training (CLIP) [31].
SimCLR RN50. ResNet50 pre-trained on ImageNet with the SimCLR loss that brings closer features of different augmentations from the same image, while pushing apart those from different images [10].
SwAV RN50. ResNet50 pre-trained on ImageNet with the SwAV mechanism that predicts the cluster assignment of a view from the representation of another one [6].
Barlow Twins RN50. ResNet50 pre-trained on ImageNet with the Barlow Twins loss that encourages the correlation of two views from the same image to be one, while discouraging that of views from different images to be zero [43].

Evaluation scenario: CIL and online CIL, where the model can only have access to the data once unless with a replay buffer. In other words, the model can not iterate over the data of the current task for multiple epochs, which is common in CIL.

Evaluation datasets: CIFAR100 with 20 tasks, CUB200, Mini-ImageNet, FGVC-Aircraft, QuickDraw

Finetuning strategy: They initialize the model with the pretrained weights and then finetune in a supervised or self-supervised approach.

Two-stage pipeline: They have two training phases. In the first one (streaming phase), the model learns from the streaming current data for 1 epoch and stores some examples in the memory. In the second phase (offline), the model learns from the samples in the buffer for 30 epochs.

Results:

Scratch

With pretraining

Split CIFAR100

Model	ER [33]	MIR [2]	GSS [4]	*LwF* [24]	iCaRL [32]	*EWC++* [8]	GDumb [30]	AGEM [9]	SCR [26]
R-RN18	9.07±1.31	8.03±0.78	6.86±0.60	8.44±0.82	14.26±0.79	1.00±0.00	9.80±0.46	3.00±0.47	**25.80±0.99**
RN18	43.69±1.67	42.02±1.53	25.59±0.45	23.40±0.12	**56.64±0.23**	5.36±0.26	46.76±0.73	4.72±0.21	51.93±0.06
Δ	+34.62	+33.99	+18.73	+14.96	**+42.38**	+4.36	+36.96	+1.72	+26.13

Split CUB200

Model	ER	MIR	GSS	*LwF*	iCaRL	*EWC++*	GDumb	AGEM	SCR
R-RN18	1.24±0.11	1.44±0.08	1.46±0.22	1.47±0.11	1.82±0.24	0.80±0.20	4.49±0.56	0.67±0.12	**5.64±0.75**
RN18	21.05±1.07	20.95±0.66	17.65±0.45	6.79±0.36	39.95±1.43	4.47±0.10	38.63±0.44	4.59±0.30	**43.03±1.80**
Δ	+19.81	+19.51	+16.19	+5.32	**+38.13**	+3.67	+34.14	+3.92	+37.39

Split Mini-ImageNet

Model	ER	MIR	GSS	*LwF*	iCaRL	*EWC++*	GDumb	AGEM	SCR
R-RN18	8.56±0.24	8.00±0.82	6.74±0.15	7.58±0.65	11.61±0.78	1.00±0.00	7.01±0.40	3.04±0.21	**33.87±1.84**
RN18	56.91±0.54	54.96±0.46	25.74±4.53	20.41±0.99	**72.40±0.52**	4.79±0.14	40.00±0.37	5.23±0.41	67.94±0.11
Δ	+48.35	+46.96	+19.00	+12.83	**+60.79**	+3.79	+29.99	+2.19	+34.07

Table 2. **Accuracy in online CIL.** Different CL algorithms benefit from a pre-trained model very differently, and the comparison results between algorithms change when they are initialized from a pre-trained model. For instance, iCaRL outperforms SCR, the best-performing model when trained from scratch, on Split CIFAR100 (56.64 vs. 51.93) and Split Mini-ImageNet (72.40 vs. 67.94). This indicates that training from scratch does not serve as a fairground for comparison between different algorithms, in addition to its poor applicability to complex datasets. R-RN18 and RN18 stand for Reduced ResNet18 trained from scratch and ImageNet pre-trained ResNet18, respectively.

Compared to from scratch, we see a huge improvement in different CL methods when pretraining is used (supervised finetuning)

Model	ER [33]	MIR [2]	GSS [4]	*LwF* [24]	iCaRL [32]	*EWC++* [8]	GDumb [30]	AGEM [9]	SCR [26]	DER++ [5]	Co^{2}L [7]
R-RN18	9.07±1.31	8.03±0.78	6.86±0.60	8.44±0.82	14.26±0.79	1.00±0.00	9.80±0.46	3.00±0.47	**25.80±0.99**	15.72±1.33	2.31±0.64
RN18	43.69±1.67	42.02±1.53	25.59±0.45	23.40±0.12	**56.64±0.23**	5.36±0.26	46.76±0.73	4.72±0.21	51.93±0.06	44.42±1.29	5.68±3.19
RN50	50.88±0.84	50.20±2.80	31.53±3.37	26.68±0.97	**59.20±0.33**	3.47±1.42	57.37±0.21	4.49±0.27	56.22±0.42	49.37±1.36	8.57±0.57
CLIP	52.31±2.66	**55.38±0.83**	25.60±4.50	37.21±2.14	26.05±12.33	—*	55.10±0.22	17.22±2.52	30.93±5.44	**53.01±0.18**	1.12±0.16
SimCLR RN50	37.04±0.48	40.01±1.86	16.32±1.52	3.40±0.17	33.76±0.84	6.39±0.82	24.63±0.84	3.87±0.32	**52.60±0.22**	15.63±0.96	1.44±0.45
SwAV RN50	38.32±0.11	40.97±0.36	15.00±0.30	3.32±0.45	24.29±1.32	3.58±3.00	20.95±1.33	3.86±0.29	**50.59±0.09**	20.10±0.88	1.18±0.26
B.T. RN50	26.15±0.62	18.18±1.60	8.38±0.23	3.70±0.16	40.77±0.92	6.65±1.06	31.56±2.01	3.95±0.31	**48.35±0.73**	5.26±0.17	1.10±0.10

*EWC++ fails to train with losses of *nan*.

Table 3. **Accuracy of different pre-trained models** *when fine-tuned in a supervised manner* on Split CIFAR100 in online CIL. In most cases, RN pre-trained on ImageNet (RN50 vs. CLIP RN50) in a supervised fashion (RN50 vs. SimCLR, SwAV, and Barlow Twins RN50) brings the largest accuracy increase. Red numbers mark pre-trained accuracy that is within/below one std. of the from-scratch counterpart, which indicates potential negative impacts. **Bold** numbers indicate the best accuracy amongst all methods with a specific model (e.g., 25.80 of SCR is the best within R-RN18). R-RN18, RN18, RN50, and CLIP stand for Reduced ResNet18 trained from scratch, ImageNet pre-trained ResNet18 and ResNet50, and CLIP pre-trained ResNet50, respectively. B.T. stands for Barlow Twins. **[Best viewed in color.]**

Different pretrained models have different influence, but overall speaking, ImageNet pretrained RN50 (i.e., **supervised fine-tuning**) yields the best results with most CL methods.

(a) Experience Replay (ER)

Fine-Tuning	From-scratch	Supervised			Self-supervised
	R-RN18	RN18	RN50	CLIP	SimCLR RN50
CIL	8.17±1.06 (63.88±1.07)	36.21±1.17 (59.05±0.82)	44.18±2.55 (50.96±2.29)	55.44±1.34 (36.36±0.98)	34.72±4.04 (20.26±1.87)
Online CIL	9.07±1.31 (52.10±2.22)	43.69±1.67 (50.22±1.82)	50.88±0.84 (42.93±0.67)	52.79±1.91 (35.51±2.90)	33.39±0.42 (20.28±0.75)

(b) Learning without Forgetting (LwF)

Fine-Tuning	From-scratch	Supervised		
	R-RN18	RN18	RN50	CLIP
CIL	13.05±0.65 (8.33±4.35)	19.18±0.86 (-4.40±1.44)	17.82±1.83 (-3.50±2.10)	35.52±1.90 (-3.97±1.25)
Online CIL	8.44±0.82 (8.83±1.31)	23.40±0.12 (-2.14±2.32)	25.98±1.34 (-3.42±1.18)	37.73±1.19 (-1.81±0.93)

Table 5. **Accuracy (Forgetting) of different models** on Split CIFAR100. (a) *Self-supervised fine-tuning* (SimCLR) demonstrates a lower forgetting compared with supervised fine-tuning (20.26 vs. 50.96 of RN50 in CIL). (a)(b) CLIP, pre-trained with image-text pairs, shows less forgetting compared with ResNets pre-trained with curated ImageNet labels. Numbers outside/inside parentheses are accuracy/forgetting, respectively. R-RN18 and RN18 stand for Reduced ResNet18 and ImageNet pre-trained ResNet18, respectively.

Self-supervised fine-tuning (SimCLR loss) has less forgetting than supervised fine-tuning.

★ **Results of the proposed two-stage baseline:**
The proposed method combines the simplest ER that exerts no regularization during training, ImageNet RN50, and the two-stage training pipeline discussed above.

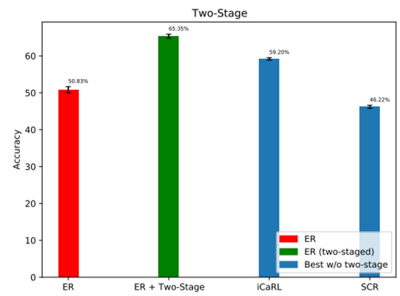

Figure 4. A simple second-staged offline training on memory data coupled with an ImageNet pre-trained ResNet50 turns a simple baseline into state of the art, suggesting the effectiveness of the proposed baseline. Note that SCR and iCaRL are the two best-performing methods when applied on the ImageNet pre-trained ResNet50. **[Best viewed in color.]**