KAIST [1]
LG AI Research [2]

DLCFT: Deep Linear Continual Fine-Tuning for General Incremental Learning

Hyounguk Shon[1], Janghyeon Lee[2], Seung Hwan Kim[2], Junmo Kim[1]

ECCV TEL AVIV 2022

Summary

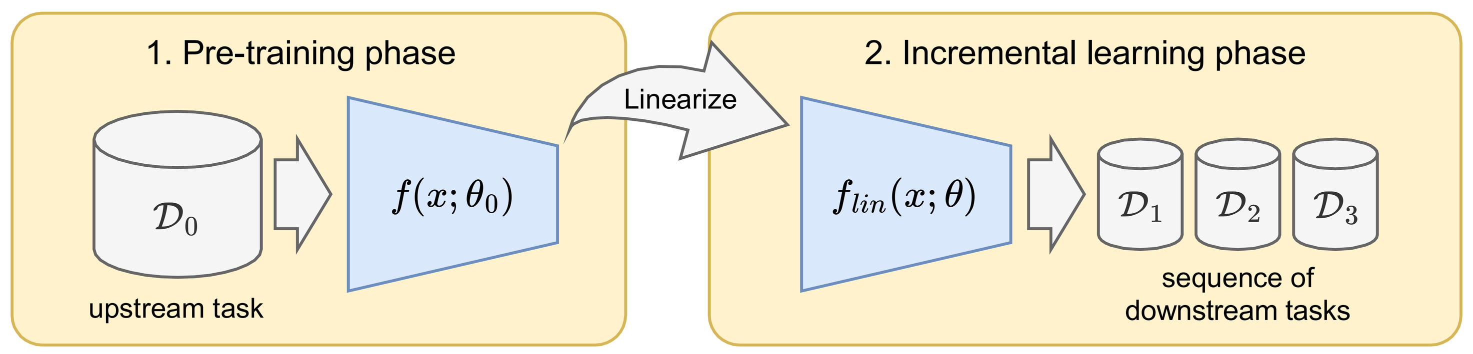
- We address Continual Fine-tuning, an alternative CL framework that leverages pre-trained representation.
- We propose a simple method that significantly boosts the performance of parameter regularization based on network linearization and MSE loss.
- Our method flexibly applies to various incremental learning settings.

Background on Continual Learning

- Goal: to learn a model incrementally through a set of sequentially arriving batches of data/tasks/classes.
- Parameter regularization methods approximate the previous task objectives using second-order Taylor approximation.

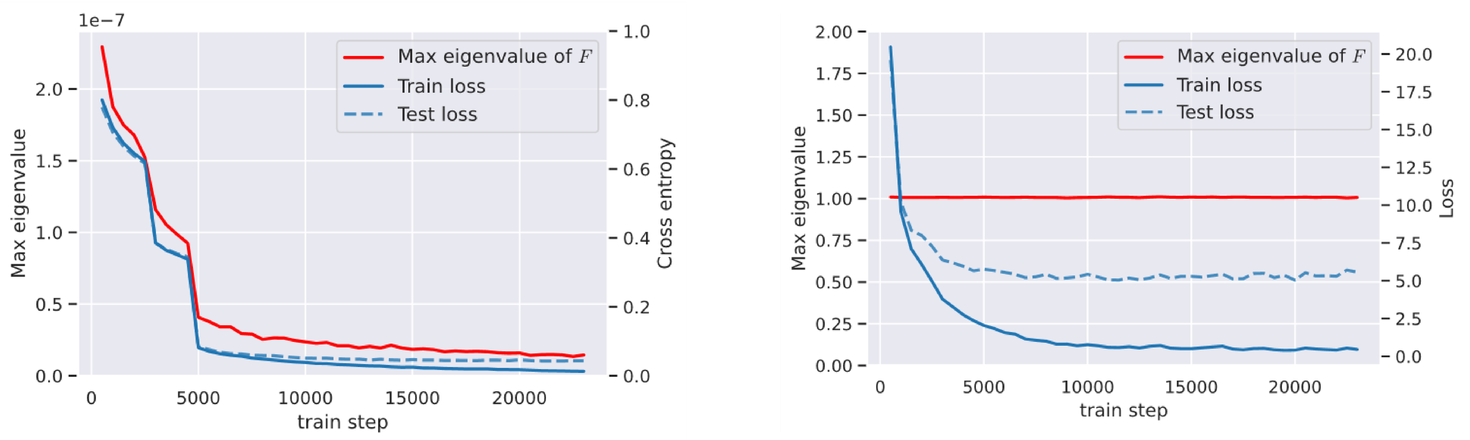
$$\frac{1}{t}\sum_{i=1}^{t}\mathop{\mathbb{E}}_{(x,y)\in\mathcal{D}_i}[\lambda_i\mathcal{L}(f(x;\theta,i),y)] = C + \frac{1}{2}(\theta-\theta_t)^\top A(\theta-\theta_t) + \mathcal{O}(\theta^3)$$

Continual Fine-tuning



Most existing methods addresses continual learning where a model is trained from scratch. Here, we explore continually fine-tuning a pre-trained model.

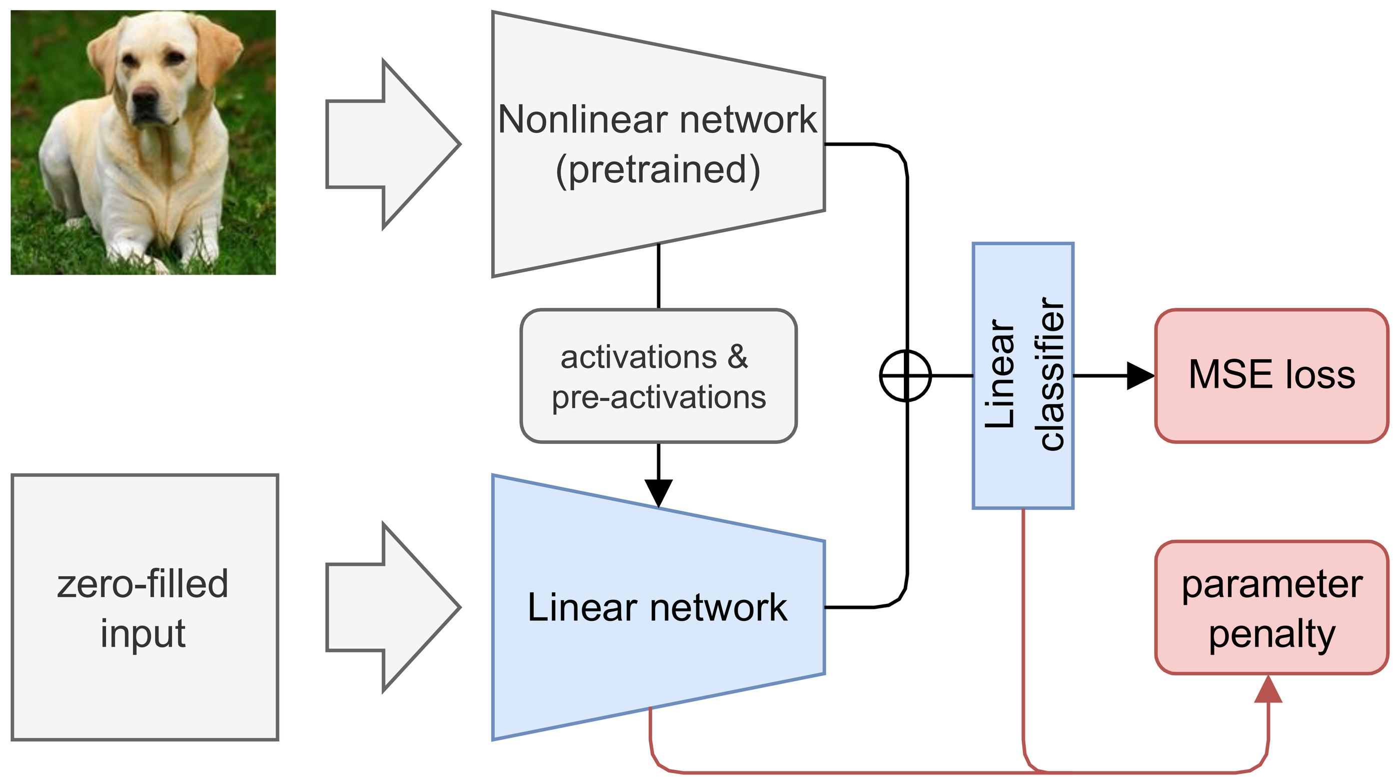
Problems of parameter regularization



(a) Curvature of SCE loss (b) Curvature of MSE loss

1. Vanishing curvature problem
Second derivatives of the softmax cross-entropy loss converges to zero as the model fits the data.
2. Higher-order error
Quadratic approximation assumes that the loss is a quadratic function of the parameters, which can lead to large error as the model learns more data.

Deep Linear Continual Fine-tuning (DLCFT)



1. To address the vanishing curvature problem, we replace softmax cross-entropy loss for MSE loss. The labels are converted into scaled one-hot vectors.

$$\ell(f(x;\theta,t),y) = \frac{1}{2}||\alpha\phi(y) - f(x;\theta,t)||^2$$

2. To address the higher-order error problem, we use linear approximation of the neural network to linearize the model.

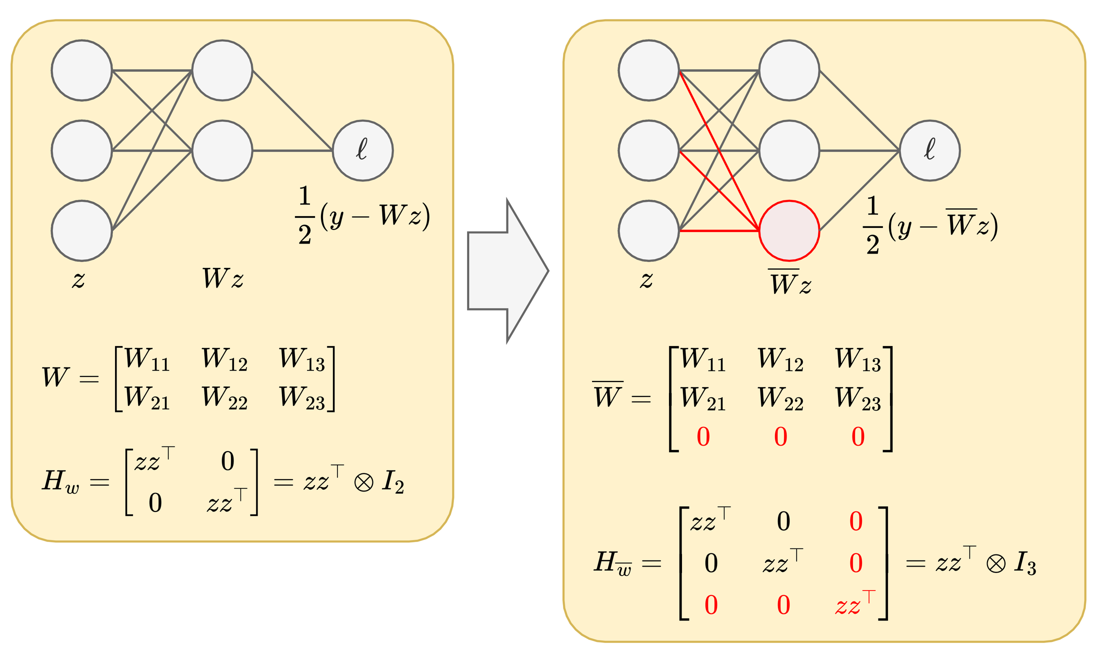
$$f(x;\theta,t) = w_t\cdot g_{lin}(x;\psi) + b_t,$$

Here, quadratic parameter regularization is the optimal continual learning policy

$$\begin{aligned}\ell(f(x;\theta,t),y) &= \frac{1}{2}||\alpha\phi(y) - f(x;\theta,t)||^2 \\ &= \frac{1}{2}(\theta-\theta_t)^\top H_\theta(\theta-\theta_t) + C\end{aligned}$$

$$\begin{aligned}\mathcal{L}_{data} &= \mathop{\mathbb{E}}_{(x,y)\sim\mathcal{D}_t}[\ell(f(x;\theta,t),y)] \\ \mathcal{L}_{reg} &= \frac{1}{2}(\theta-\theta_{t-1})^\top H(\theta-\theta_{t-1}) \\ \mathcal{L} &= \frac{1}{t}\mathcal{L}_{data} + \frac{t-1}{t}\mathcal{L}_{reg},\end{aligned}$$

Classifier regularization for class-IL problem



$$\begin{aligned}\frac{\partial^2\mathcal{L}}{\partial\overline{w}\partial\overline{w}^\top} &= \frac{\partial}{\partial\overline{w}}\mathbb{E}[(y-\overline{W}z)z^\top] \\ &= \mathbb{E}[zz^\top\otimes I] \\ &= \mathbb{E}[zz^\top]\otimes I.\end{aligned}$$

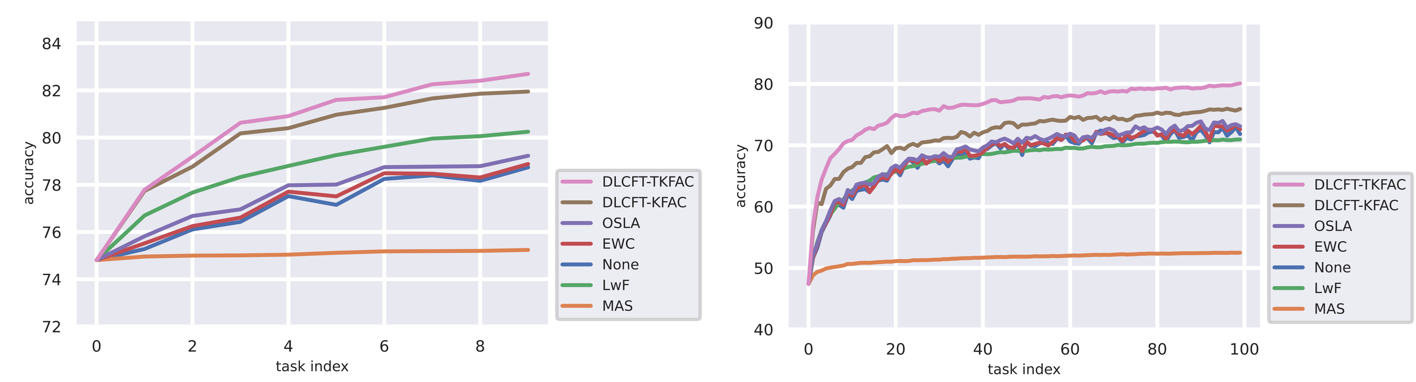
Additionally, a small buffer is used to replay examples.

$$\mathcal{L}_{reg} = \frac{\lambda}{2}\cdot(\theta-\theta_{t-1})^\top H(\theta-\theta_{t-1}) + (1-\lambda)\cdot\mathop{\mathbb{E}}_{\mathcal{M}}[\ell(f(x;\theta,t),y)]$$

We can obtain the curvature for the unseen weights without requiring the previous task data.

Experiments

Data-incremental learning



(a) Seq-CIFAR-100 (10 tasks) (b) Seq-CIFAR-100 (100 tasks)

Dataset	**Seq-CIFAR-100**		**Seq-MIT-67**
Sequence length	10	100	4
None	78.74	71.83	63.48
LwF [24]	80.25	70.95	67.21
EWC [18]	78.88	72.61	63.68
MAS [2]	75.24	52.50	62.49
OSLA [38]	79.23	73.10	64.08
DLCFT† (Ours)	**81.95**	**75.92**	**70.55**
DLCFT‡ (Ours)	**82.70**	**80.07**	**70.52**
Joint	83.57		74.40

Our method scales well to learning a very long sequence of data

Task- / Class-incremental learning

Buffer size	Method	Task-IL	Class-IL
0	LwF [24]	92.16	-
	EWC [18]	77.44	-
	OSLA [38]	81.03	-
	DLCFT (Ours)	**95.79**	-
500	ER [37]	79.14	43.52
	DER [4]	91.47	58.07
	DER++ [4]	91.56	53.29
	DLCFT (Ours)	-	**59.98**

Our method improves performance when combined with experience replay

Ablations

			Data-IL		**Class-IL**
Curvature	Linear	Loss	10 tasks	100 tasks	10 tasks
K-FAC [30]	✗	SCE	79.23	73.51	44.93
	✓	MSE	**81.95**	**75.92**	**59.86**
TK-FAC [10]	✗	SCE	79.58	73.18	44.66
	✓	MSE	**82.70**	**80.07**	**59.98**

Combination of linearization and MSE loss is critical to performance.