Parameter-Efficient & Memory-Efficient Fine-tuning

Author: Marco Pichler
Assistant Supervisor: Richard Freinschlag

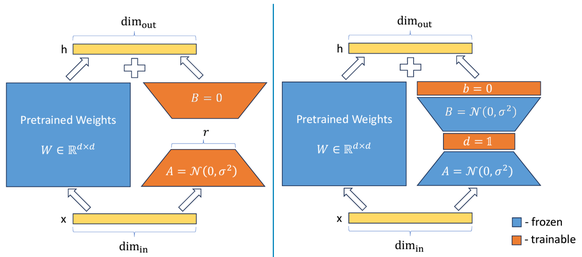
JKU MACHINE LEARNING

Motivation

Parameter-efficient and memory-efficient fine-tuning enables the adaptation of pre-trained deep learning models with:

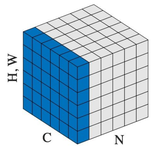
- **Reduced computational costs**
- **Lower memory demand**

LoRA and VeRA [1] [2]

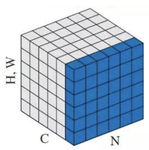


Normalization

Layer Normalization [3]



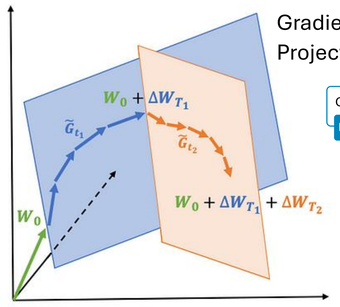
Batch Normalization [4]

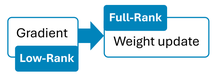


$$y = \frac{x - \mathrm{E}[x]}{\sqrt{\mathrm{Var}[x] + \epsilon}} * \gamma + \beta$$

GaLore

Gradient Low-Rank Projection [5]

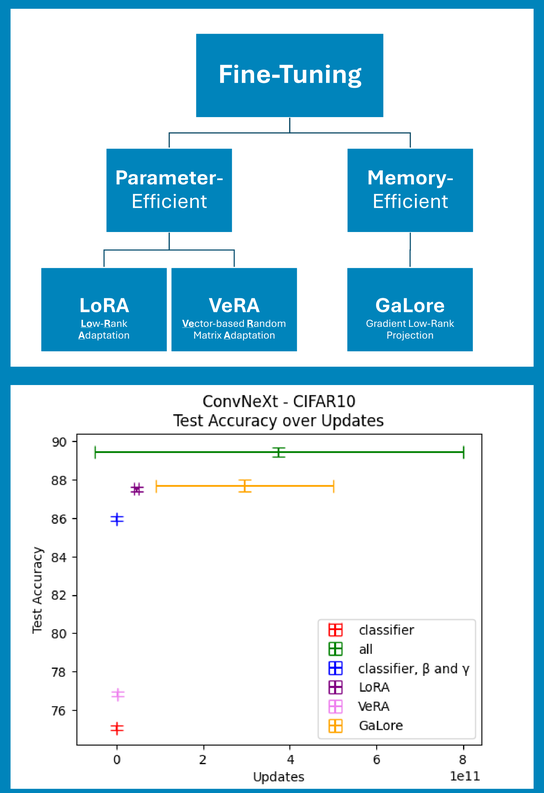


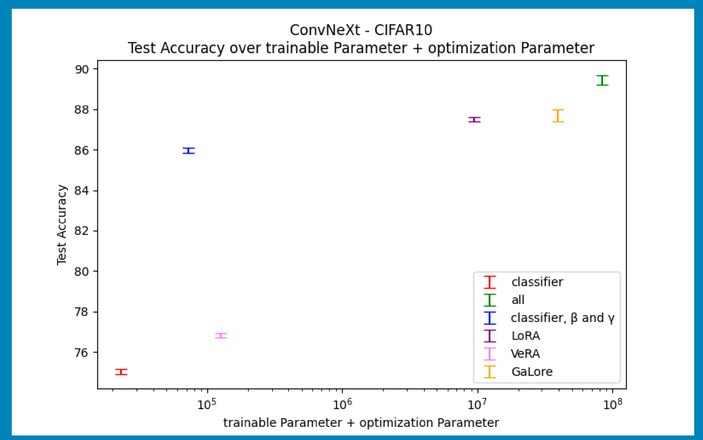


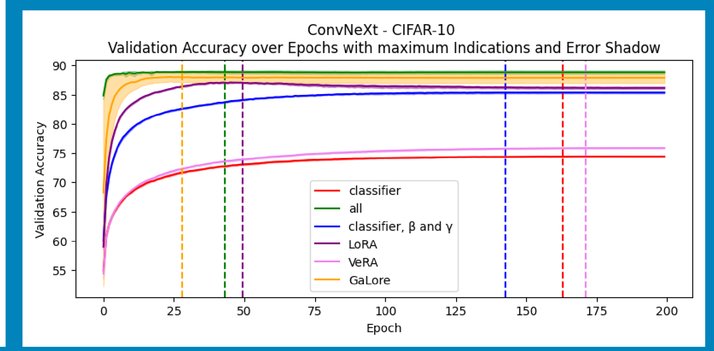
$$W_T = W_0 + \eta \sum_{t=0}^{T-1} \tilde{G}_t$$

$$\tilde{G}_t = P_t \rho_t(P_t^\top G_t Q_t) Q_t^\top$$

$P_t \in \mathbb{R}^{m \times r}$ and $Q_t \in \mathbb{R}^{n \times r}$ are projection matrices.





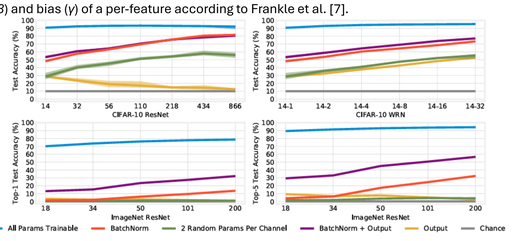


The study conducted by Lu et al. [6] investigates how pretrained transformers can be fine-tuned with a minimal number of parameters. The approach is to freeze both the feedforward layers of the residual blocks and the self-attention and to fine-tune only the input layer (embedding), the output layer (readout) and the **LayerNorm** parameters (gain γ and bias β) of the pre-trained transformer. The inspected transformer is the GPT-2.

Results of the study conducted by Lu et al. [6] show how combinations of fine-tuning parameters affects the performance.

Task	output	output + input	output + positions	output + layernorm
Bit Memory	76 %	**98 %**	93 %	94 %
Bit XOR	56 %	72 %	84 %	**98 %**
ListOps	15 %	17 %	35 %	**36 %**
MNIST	23 %	85 %	93 %	**96 %**
CIFAR-10	25 %	53 %	38 %	**54 %**
CIFAR-10 LRA	17 %	22 %	30 %	**39 %**
Homology	2 %	8 %	8 %	**9 %**

The research by Frankle et al. [7] investigates how random initialized CNNs can be fine-tuned by only training the parameters of **BatchNorm**. The used CNNs are variations of ResNet. The expressive power of the BatchNorm parameters β and γ is greater than other parameters. This results from the special position which allows them to be a coefficient (β) and bias (γ) of a per-feature according to Frankle et al. [7].



The research by Dong et al. [8] shows how fine-tuning ViT-B/16 (pre-trained on ImageNet-21k) with **LoRA** affects the performance on a variety of image classification tasks.

Group	Dataset	Full fine-tuning	LoRA
Natural	CIFAR-100	68.9 %	**67.1 %**
	Caltech101	87.7 %	**91.4 %**
	DTD	64.3 %	**69.4 %**
	Flowers102	97.2 %	**98.8 %**
	Pets	86.9 %	**90.4 %**
	SVNH	**87.4 %**	85.3 %
	Sun397	38.8 %	**54.0 %**
	Mean	75.9 %	**79.5 %**
Specialized	Camelyon	79.7 %	**84.9 %**
	EuroSAT	**95.7 %**	95.3 %
	Resisc45	84.2 %	**84.4 %**
	Retinopathy	**73.9 %**	73.6 %
	Mean	83.4 %	**84.6 %**
Structured	Clevr-Count	56.3 %	**82.9 %**
	Clevr-Dist	58.6 %	**69.2 %**
	DMLab	41.7 %	**49.8 %**
	KITTI-Dist	65.5 %	**78.5 %**
	dSpr-Loc	57.5 %	**75.7 %**
	dSpr-Ori	46.7 %	**47.1 %**
	sNORB-Azim	25.7 %	**31.0 %**
	sNORB-Ele	29.1 %	**44.0 %**
	Mean	47.6 %	**59.8 %**
	Mean Total	65.6 %	**72.3 %**
	Params.(M)	85.8 M	0.29 M

This thesis focuses on image classification and therefore, the study by Dong et al. [6] is of interest since LoRA was used to fine-tune a Vision Transformer (ViT) for image classification.

LoRA performes better than the full fine-tuning variant in **16 out of 19** datasets. In the **Natural** dataset group, LoRA performs on average **3.6%** points better than the full fine-tuning variant, and for the **Specialized** dataset group the gap is only **1.2%** points. The difference between the average performance on the **Structured** dataset group, is greatest, at **12.2%** points.

On **average across all datasets**, LoRA performed **6.7% points** better than the full fine-tuning variant, despite LoRA only having **0.34%** of the amount of **learnable parameters** as the full fine-tuning variant.

The Natural dataset group contains daily life images. The Specialized dataset group contains images obtained with special devices, including remote sensing and medical images. The Structured dataset group contains artificial images from visual simulated environments.

The study conducted by George et al. [9] proposes a new variant of **GaLore**, which can handle higher order tensor weights. To use GaLore on a weight tensor so that the SVD can be calculated for projection into a low-rank space, the weights must be in form of a matrix. Most of the information can be preserved when the reshaping originates from a weight matrix, which resembles a vector-mapping operator (e.g. Linear Layer). If the weight matrix has a higher dimension, originated from a multi-dimensional mapping operator, important information may be lost in the reshaping process when the rank of the matrix is flattened into fewer dimensions, as the study conducted by George et al. [9] shows. Such a multi-dimensional mapping operator is for example a convolutional layer. [9] EfficientNet has no Linear Layer apart from the classifier.

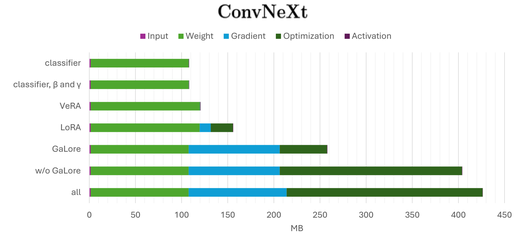
Results

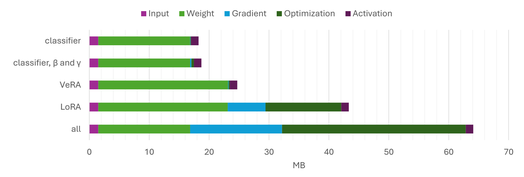
ConvNeXt	CIFAR-10	CIFAR-100	MedMNIST
classifier	75.060	50.980	79.525
all	**89.438**	66.434	**88.998**
classifier, β and γ	85.964	66.334	86.402
LoRA	87.516	**66.924**	86.992
VeRA	76.836	52.814	79.783
GaLore	87.696	63.632	88.043
w/o GaLore	89.238	66.098	88.873

EfficientNet	CIFAR-10	CIFAR-100	MedMNIST
classifier	40.158	19.866	67.876
all	**81.256**	**55.638**	**85.128**
classifier, β and γ	62.720	37.298	77.535
LoRA	72.660	42.122	78.573
VeRA	40.550	19.874	66.340

The MedMNIST column shows the average results of the DermaMNIST, BloodMNIST, OrganCMNIST, BreastMNIST and PneumoniaMNIST datasets.

Memory Usage





Conclusion

- **LoRA** is highly accurate, converges faster than Norm-fine-tuning and works better with Linear Layers than with ConvLayer but adds parameters.
- **VeRa** has a lower accuracy than LoRA, but memory usage is reduced despite the additional parameters.
- **GaLore** reduces memory usage and converges with lowest epochs but can only be applied on Linear Layers.
- Norm-fine-tuning is the most memory-efficient for ConvNeXt (**LayerNorm**) among the truly useful variants but needs more epochs to converge.
- Fine-tuning Normalization does not work well for EfficientNet (**BatchNorm**).
- Fine-tuning only the **classifier** has the lowest memory consumption but has also the worst accuracy and converges slowly and therefore, is not a useful variant.
- **Norm-fine-tuning** is the best compromise for **ConvNeXt** and **LoRA** is the best compromise for **EfficientNet** when memory is precious.

[1] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.
[2] Dawid J Kopiczko, Tijmen Blankevoort, and Yuki M Asano. 2023. Vera: Vector-based random matrix adaptation. arXiv preprint arXiv:2310.11454.
[3] Jimmy Lei Ba. 2016. Layer normalization. arXiv preprint arXiv:1607.06450.
[4] Sergey Ioffe. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167.
[5] Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. 2024. Galore: Memory-efficient llm training by gradient low-rank projection. arXiv preprint arXiv:2403.03507.
[6] Kevin Lu, Aditya Grover, Pieter Abbeel, Igor Mordatch, and Michael Laskin. 2021. Pretrained transformers as universal computation engines. CoRR abs/2103.05247 (2021). arXiv preprint arXiv:2103.05247.
[7] Jonathan Frankle, David J Schwab, and Ari S Morcos. 2020. Training batchnorm and only batchnorm: On the expressive power of random features in cnns. arXiv preprint arXiv:2003.00152.
[8] Wei Dong, Xing Zhang, Bihui Chen, Dawei Yan, Zhijun Lin, Qingsen Yan, Peng Wang, and Yang Yang. 2024. Low-Rank Rescaled Vision Transformer Fine-Tuning: A Residual Design Approach. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16101–16110.
[9] Robert Joseph George, David Pitt, Jiawei Zhao, Jean Kossaifi, Cheng Luo, Yuan- dong Tian, and Anima Anandkumar. 2025. Tensor-GaLore: Memory-Efficient Training via Gradient Tensor Decomposition. arXiv preprint arXiv:2501.02379.