Geometry-aware training of factorized layers in tensor Tucker format

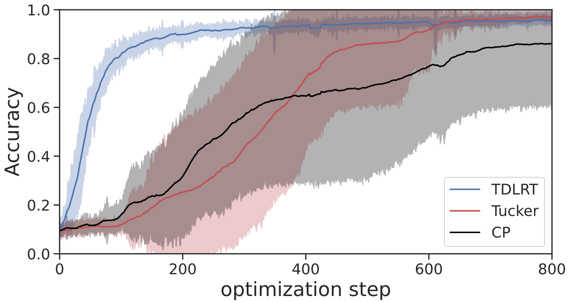
Emanuele Zangrando[1], Steffen Schotthöfer[2], Gianluca Ceruti[3], Jonas Kusch[4], Francesco Tudisco[1,5]

[1] Gran Sasso Science Institute
[2] Oak Ridge National Laboratory
[3] University of Innsbruck
[4] Norwegian University of Life Sciences
[5] University of Edinburgh

NEURAL INFORMATION PROCESSING SYSTEMS

Problem

- Train or fine-tune neural network with tensor layers (e.g. convolutions) in a low-rank memory efficient format
- train networks in a rank-adaptive fashion
- Avoid instabilities of factorized gradient descent

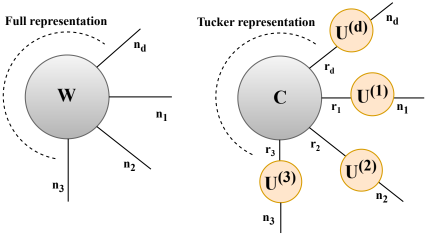


Original dynamics

$W \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, full-training consists in discretizing the gradient flow

$$\dot{W} = -\nabla L(W(t))$$

space: $O(\prod_{i=1}^{d} n_i)$



Tucker decomposed layer:

$$W_{i_1,\dots,i_d} = C^{j_1,\dots,j_d} U^{(1)}_{j_1,i_i} \dots U^{(d)}_{j_d,i_d}$$

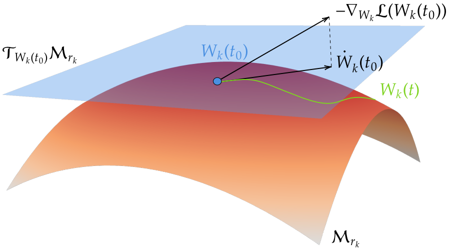
Block-dynamics:

$$\dot{U}^{(i)} = -\nabla_{U^{(i)}} L(C \times_j U^{(j)})$$
$$\dot{C} = -\nabla_C L(C \times_j U^{(j)})$$

Low-rank dynamics approximation

$$\mathcal{M}_\rho := \{W \in \mathbb{R}^{n_1\times\cdots\times n_d} \,|\, \mathrm{rank}(\mathrm{Mat_i(W)}) = r_i\}, \quad \rho = (r_1, \ldots, r_d)$$

$$\dot{W} = -P_{T_W\mathcal{M}_\rho}\nabla L(W)$$



Best 1^{st} order local approximation of the original dynamics

$$\dot{U}^{(i)} = -(I - U^{(i)}U^{(i)\top})\mathrm{Mat_i}(\nabla L(W) \mathop{\times}_{j\neq i} U^{(j)\top})\mathrm{Mat_i}(C)^\dagger$$

$$\dot{C} = -\nabla L(W) \mathop{\times}_{j=1}^{d} U^{(j)\top}$$

space: $O(\prod_{i=1}^d n_i r_i + \prod_{i=1}^d r_i)$

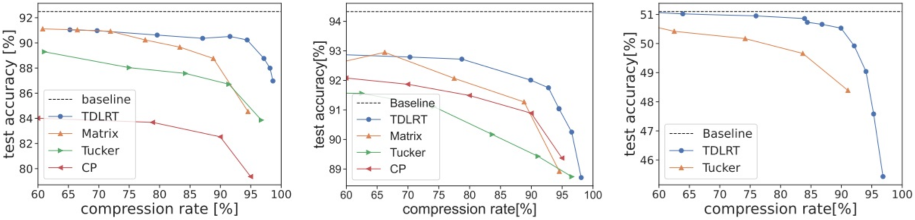
Results

- Theoretical guarantees of approximation of the original problem and descent of the loss, together with convergence to critical points in the stochastic setting

$$\|W(t) - C(t) \times_j U_j(t)\| \leq c_1\varepsilon + c_2\lambda + c_3\tau/\lambda$$
$$\liminf_{t\to\infty} \mathbb{E}\|\nabla L(W(t+1)) \times_j U_j(t)U_j(t)^\top\|^2 = 0$$

- Efficient and robust integration of the projected system to optimize the neural net (constants c_i do not depend on the singular values!)
- Variety of different experiments ranging from low-rank training from scratch to fine-tuning pretrained models

Experiments



(b) VGG16 Cifar10 (c) ResNet18 Cifar10 (d) ResNet18 Tiny-Imagenet

Figure: Compressed training from scratch

Experiments

Table 2: Fine-tuning performance metrics on Deberta V3 Glue benchmark (left) and on Stable diffusion Dreambooth (right).

GLUE # params	**LoRA** 1.33M (rank 8)	**TDRLT**(Ours) 0.9M ($\tau = 0.15$)
CoLa (Corr.)	0.6759	0.7065
MRPC (Acc.)	0.8971	0.9052
QQP (Acc.)	0.9131	0.9215
RTE (Acc.)	0.8535	0.8713
SST2 (Acc.)	0.9484	0.9594

method	**loss**	**# params**
LoRA ($r = 8$)	0.260	5 M
LoRA ($r = 5$)	0.269	3 M
LoRA ($r = 3$)	0.274	1.8 M
Ours ($\tau = 0.02$)	0.2635	1.8 M
Ours ($\tau = 0.1$)	0.272	1.5 M

Figure: Fine-tuning with low-rank tensor and matrix adapters

Poster

We are happy to welcome you at our poster session for further discussion

Friday 13 December, 4:30-7:30 pm local time