dEBORA: Efficient Bilevel Optimization-based low-Rank Adaptation

Emanuele Zangrando[1], Sara Venturini[2], Francesco Rinaldi[3], Francesco Tudisco[1,4,5]

[1] Gran Sasso Science Institute
[2] MOBS Lab, Northeastern University
[3] University of Padova
[4] University of Edinburgh
[5] Miniml.AI

Formulation

- Problem: fine-tune neural network with matrix or tensor layers using LoRA, and dynamically update the rank depending on the problem

Defined the tensor adapter through a CP like decomposition

$$\Psi(\mathcal{B}, s) = \sum_{i=1}^{r} s_i u_i^{(1)} \otimes \cdots \otimes u_i^{(d)}, \quad u_i^{(j)} \in \mathbb{R}^{n_j}$$

We can naturally formulate model compression/ fine-tuning as a bilevel optimization problem

$$\begin{aligned} &\text{(upper-level)} && \min_{s \in \mathbb{R}^r : s \geq 0, \|s\|_1 \leq \tau} && f_1(\mathcal{B}^*(s), s) \\ &\text{(lower-level)} && \text{s.t.} \quad \mathcal{B}^*(s) \in \underset{\mathcal{B} \in \mathcal{V} \subseteq \mathbb{R}^{n_1 \times r} \times \cdots \times \mathbb{R}^{n_d \times r}}{\arg\min} && f_2(\mathcal{B}, s), \end{aligned} \tag{1}$$

where $f_i(\mathcal{B}, s) = \sum_{(x,y) \in \mathcal{D}_i} \ell(\mathcal{W}_0 + \Psi(\mathcal{B}, s); x, y), \quad \mathcal{B} = (u_i^{(j)})_{i,j}$.

Structure of constraint set: Frank-Wolfe

The constraint set

$$\mathcal{S} = \left\{ s \in \mathbb{R}^r \mid s \geq 0, \|s\|_1 \leq \tau \right\}$$

is a simplex, which makes the method amenable for Frank-Wolfe, which has the advantage of being projection-free.

Problem: The LMO of Frank-Wolfe requires calculating the hypergradient $\nabla_s f_1(\mathcal{B}^*(s), s) = \partial_{\mathcal{B}} f_2^* \partial_s \mathcal{B}^*(s) + \partial_s f_2^*$, which requires solving a linear system in $\partial_s \mathcal{B}^*(s)$

$$\partial^2_{\mathcal{B}} f_2^* \partial_s \mathcal{B}^* = -\partial_s \partial_{\mathcal{B}} f_2^*$$

Results

Efficient hypergradient approximation $G(s)$ as a function of $\mathcal{B}^*(s), \nabla_{\mathcal{B}} f_1^*$

- Allows performing the Frank-Wolfe step without the need of implicit differentiation or other techniques;
- Error estimate $\|G(s) - \nabla_s f_1^*(s)\| \leq C\|\nabla^2 L_2(\Psi(\mathcal{B}^*(s), s))\|$;
- Convergence guarantees in the stochastic noisy hypergradient setting, allowing to estimate G also on a minibatch;
- Identifiability guarantees of the optimal face of the L^1 simplex in a finite number of steps, allowing to continue with the optimal rank

Experiments

Table: DeBERTaV3-base fine-tuning on GLUE benchmark.

Method	# Params (Acc)	MNLI (Acc)	SST-2 (Mcc)	CoLA (Acc/F1)	QQP (Acc)	QNLI (Acc)	RTE (Acc)	MRPC (Corr)	STS-B
Full FT	184M	89.90	95.63	69.19	92.40/89.80	94.03	83.75	89.46	91.60
HAdapter	1.22M	90.13	95.53	68.64	89.27	94.11	84.48	89.95	91.48
PAdapter	1.18M	90.33	95.61	68.77	89.40	**94.29**	85.20	89.46	**91.54**
LoRA $r = 8$	1.33M	90.29	95.29	68.57	90.62/90.61	93.91	85.5	89.75	89.10
AdaLoRA	1.27M	**90.44**	95.64	68.76	90.59/90.65	94.11	**86.00**	89.44	91.41
$\tau = 16$	**0.4M**	90.01	95.29	68.72	**91.88/89.20**	93.42	83.75	90.16	90.84
(Oblique) $\tau = 16$	**1.03M**	90.0	94.83	**69.15**	88.62/85.09	87.33	83.03	**91.42**	91.24
(Stiefel) $\tau = 16$	**0.8M**	89.79	**95.65**	68.39	89.74/86.58	93.83	84.12	91.18	**91.54**

Poster

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

Saturday 26 April, 3:00-5:30 pm local time