On the Effectiveness of Parameter-Efficient Fine-Tuning

Zihao Fu,[1] Haoran Yang,[2] Anthony Man-Cho So,[2] Wai Lam[2] Lidong Bing,[3] Nigel Collier[1]

[1]Language Technology Lab, University of Cambridge, UK, [2]The Chinese University of Hong Kong, [3]DAMO Academy, Alibaba Group

We Answer Two Questions for Parameter-Efficient Models

- Q1. why do parameter-efficient models (such as Adapters, LoRA, Bitfit, etc.) achieve promising results?
  - Our Answer: The sparsity itself improves the model stability and generalization capability.
- Q2. How to choose the tunable parameters?
  - Our Answer: Use our Second-order Approximation Method (SAM).

Main Contributions

- We propose a new categorization scheme for existing parameter-efficient methods and generalize most of these methods with a unified view called the sparse fine-tuned model.
- We conduct a theoretical analysis of the parameter-efficient models' stability and generalization.
- We propose a novel SAM model to choose the suitable parameters to optimize.
- We conduct extensive experiments to verify our theoretical analysis and the SAM model.

Unified View of Parameter Efficient Fine-tuning

All the following models are p-sparse fine-tuned models.

- Random Approaches
  - **Random** model: Randomly selecting the parameters with respect to a given sparsity ratio.
  - **Mixout** (Lee et al., 2019) : Reset a portion of the fine-tuned model's parameters to the pre-trained parameters with respect to a ratio.
- Rule-Based Approaches
  - **BitFit** (Zaken et al., 2021) : Only fine-tunes the bias terms.
  - **MagPruning** (Han et al., 2015a,b; Lee et al., 2021) : Tunes the parameters with large norms.
  - **Adapter** (Houlsby et al., 2019; Pfeiffer et al., 2020) : Adapter proposes to add an adapter layer inside the transformer layer. It can be viewed as fine-tuning an equivalent model shown in Fig. **??** (a).
  - **LoRA** (Hu et al., 2022; Karimi Mahabadi et al., 2021) : LoRA proposes to add a new vector calculated by recovering a hidden vector from a lower dimension space. The model is illustrated in Fig. **??** (b).
- Projection-Based Approaches
  - **DiffPruning** (Mallya et al., 2018; Sanh et al., 2020) : Model the parameter selection mask as a Bernoulli random variable and optimize it with the reparametrization method. It then projects the mask onto its feasible region and does the optimization alternately.
  - **ChildPruning** (Xu et al., 2021; Mostafa and Wang, 2019) : Iteratively trains the full model parameters and calculates the projected mask.

Definition 1 (p-Sparse Fine-tuned Model). Given a pre-trained model $\mathcal{M}^0$ with parameters θ^0, if a fine-tuned model $\mathcal{M}$ with parameters θ has the same structure as $\mathcal{M}^0$ such that $\|\theta - \theta^0\|_0 \leq p\dim(\theta), p \in (0,1)$, we say the model $\mathcal{M}$ is a p-sparse fine-tuned model with the sparsity p.

- We first give the definition of p-sparse fine-tuned model.
- We show most of the existing parameter-efficient models are p-sparse fine-tuned models.
- We will conduct our theoretical analysis on p-sparse fine-tuned model which can naturally explain most of the existing models.

Theoretical Analysis of the Sparse Fine-tuned Model

We denote the original training data as $S = \{z_1, \cdots, z_n\}$ and the dataset without one sample as $S^i = S \backslash z_i = \{z_1, \cdots, z_{i-1}, z_{i+1}, \cdots, z_n\}$, where z_i is the ith training sample. We also define $i \sim U(n)$ as a sampling procedure from a uniform distribution with n samples. $\mathcal{A}(S)$ is defined as model parameters obtained by running algorithm $\mathcal{A}$ on data S.

Definition 2 (Pointwise Hypothesis Stability, (Bousquet and Elisseeff, 2002)). We say that a learning algorithm $\mathcal{A}$ has **pointwise hypothesis stability** ϵ with respect to a loss function ℓ, if

$$\mathbb{E}_{S,i\sim U(n)}[|\ell(\mathcal{A}(S^i), z_i) - \ell(\mathcal{A}(S), z_i)|] \leq \epsilon. \quad (1)$$

Theorem 1 (Stability). *If the loss function ℓ is $\rho-$Lipschitz, $\mathcal{A}(S^i)$ is close to $\mathcal{A}(S)$, the Hessian matrix $\nabla^2\mathcal{L}(\mathcal{A}(S))$ at $\mathcal{A}(S)$ is positive-semidefinite with a singular value decomposition $U\,\mathrm{diag}(\Lambda)U^{-1}$, $\Lambda = \{\Lambda_1, \cdots, \Lambda_m\}$ and $\Lambda_{min} = \min\{\Lambda_1, \cdots, \Lambda_m\}$, then the expectation of the loss $\mathbb{E}_M L_R$ has a pointwise hypothesis stability as:*

$$\mathbb{E}_{S,i\sim U(n)}[|\ell(\mathcal{A}(S^i), z_i) - \ell(\mathcal{A}(S), z_i)|] \leq \frac{2\rho^2}{(\Lambda_{min} + 2(1-p))n}. \quad (2)$$

- As the sparsity parameter p decreases, the upper bound also decreases.
- Sparse models imply better stability.
- If p is too small enough, the upper bound will not change significantly. Now, the denominator is dominated by Λ_{min}.

Theorem 2 (Generalization). *We denote the generalization error as $R(\mathcal{A}, S) = \mathbb{E}_z\,\ell(\mathcal{A}(S), z)$ and the empirical error as $\hat{R}(\mathcal{A}, S) = \frac{1}{n}\sum_{i=1}^{n}\ell(\mathcal{A}(S), z)$. Then, for some constant C, we have with probability $1-\delta$,*

$$R(\mathcal{A}, S) \leq \hat{R}(\mathcal{A}, S) + \sqrt{\frac{C^2 + \frac{24C\rho^2}{\Lambda_{min}+2(1-p)}}{2n\delta}}. \quad (3)$$

- Generalization error upper bound becomes smaller as the fine-tuned parameters become sparser.
- If p is small enough, as p continue to decrease, the training error $\hat{R}(\mathcal{A}, S)$ will possibly increase when the tunable parameters are not enough to fit the data.
- Consequently, as the sparsity decreases, the generalization error will first decrease and then increase.

Second-order Approximation Method

With the above setup, we can get the optimal parameter mask M for Problem (**??**) based on the following theorem:

Theorem 3 (SAM). *If $\hat{M}_{ii} = \mathbb{1}(\sum_{j=1}^{m}\mathbb{1}(|\frac{\nabla\mathcal{L}(\theta^0)_i^2}{h_i}| > |\frac{\nabla\mathcal{L}(\theta^0)_j^2}{h_j}|) \geq m - \lfloor mp \rfloor)$, where $\nabla\mathcal{L}(\theta^0)_i$ is the ith element of the gradient vector $\nabla\mathcal{L}(\theta^0)$, then*

$$\inf_{\Delta\theta}\mathcal{L}(\theta^0 + \hat{M}\Delta\theta) \leq \inf_{\substack{\Delta\theta, \|M\|_0 = \lfloor mp \rfloor; \\ M_{ij}=0, \forall i\neq j; M_{ii}\in\{0,1\}}}\mathcal{L}(\theta^0 + M\Delta\theta). \quad (4)$$

- Selecting features according to Theorem 3 achieves the minimal value in Problem (**??**).
- The remaining problem is how to calculate the diagonal of the Hessian matrix.
- It is still very complex.

To solve this problem, instead of minimizing the target function in Problem (**??**), we propose to optimize its upper bound.

$$\begin{aligned} &\min_{\Delta\theta}\mathcal{L}(\theta^0) + \nabla\mathcal{L}(\theta^0)^{\mathrm{T}}M\Delta\theta + \frac{1}{2}(M\Delta\theta)^{\mathrm{T}}DM\Delta\theta \\ &s.t. \quad \|M\|_0 = \lfloor mp \rfloor; \quad M_{ij} = 0, \forall i \neq j; \quad M_{ii} \in \{0,1\}, \end{aligned} \quad (5)$$

where $D = \mathrm{diag}\{|\lambda_{max}|, |\lambda_{max}|, \cdots, |\lambda_{max}|\}$ and λ_{max} is the maximal eigenvalue of H.

- Second-order Approximation Method
  - (1) Get the gradient $\nabla\mathcal{L}(\theta^0)_i$ for the ith parameter θ_i.
  - (2) Calculate $|\nabla\mathcal{L}(\theta^0)_i^2|$ and take the top $\lfloor mp \rfloor$ parameters to optimize.
  - (3) Optimize selected parameters.

Experiments

	CoLA	STS-B	MRPC	RTE	CB	COPA	WSC	AVG
FullTuning	58.36±1.74	89.80±0.52	**89.55**$_{[1]}$±0.81	76.03±2.14	88.93$_{[2]}$±2.37$_{[2]}$	67.70±4.41	53.10±6.18	74.78±2.60
Random	58.35±1.05$_{[2]}$	89.81±**0.11**$_{[1]}$	88.73±0.80	72.71±3.23	**90.54**$_{[1]}$±3.39	68.80±2.64	52.88±5.97	74.55±2.46
MixOut	58.66±1.96	90.15$_{[3]}$±0.17	88.69±0.60$_{[3]}$	**77.55**$_{[1]}$±**1.64**$_{[1]}$	86.51±4.13	71.30±4.84	52.98±6.78	75.12$_{[3]}$±2.88
Bitfit	56.67±1.45	90.12±0.14$_{[3]}$	87.35±0.58$_{[3]}$	72.74±2.47	86.96±3.20	71.20±3.79	55.10±5.39	74.31±2.43
MagPruning	56.57±2.47	90.30$_{[2]}$±0.14$_{[3]}$	88.09±0.79	73.53±1.84$_{[3]}$	81.25±3.50	71.50$_{[3]}$±2.46$_{[2]}$	55.67±**2.73**$_{[1]}$	73.85±1.99$_{[3]}$
Adapter	**62.11**$_{[1]}$±1.22$_{[3]}$	90.05±0.13$_{[3]}$	89.29$_{[3]}$±0.60$_{[3]}$	76.93$_{[3]}$±2.05	87.32±4.62	69.50±2.54$_{[3]}$	57.02$_{[2]}$±5.27	76.03$_{[2]}$±2.35
LoRA	60.88$_{[3]}$±1.48	87.19±0.51	89.53$_{[2]}$±0.62	76.97$_{[2]}$±1.92	84.64±3.76	69.70±2.83	56.84$_{[3]}$±4.52	75.11±2.24$_{[3]}$
DiffPruning	58.53±1.49	89.59±0.34	78.79±6.09	69.93±7.87	86.25±2.65$_{[3]}$	72.10$_{[2]}$±2.91	53.37±3.60$_{[3]}$	72.65±3.57
ChildPruning	60.00±1.29	89.97±1.51	87.19±3.86	75.76±4.38	86.61±3.22	69.40±4.00	55.59±3.81	74.93±3.15
SAM	60.89$_{[2]}$±**0.96**$_{[1]}$	**90.59**$_{[1]}$±0.14$_{[3]}$	88.84±**0.49**$_{[1]}$	76.79±1.72$_{[2]}$	88.93$_{[2]}$±**1.75**$_{[1]}$	**74.30**$_{[1]}$±**2.45**$_{[1]}$	**59.52**$_{[1]}$±3.08$_{[2]}$	**77.12**$_{[1]}$±**1.51**$_{[1]}$

Table 1. Main experiment. We run each experiment 10 times with different random seeds and report means and standard deviations. The number in the bracket is the rank for the scores in the corresponding column.

- Most of the parameter-efficient models achieve better performance than the FullTuning model.
- Most of the parameter-efficient models are more stable than the FullTuning model.
- Random model is more stable than the FullTuning model on average.
- SAM model outperforms several baseline models in several tasks and it ranks in the top 3 of most tasks.

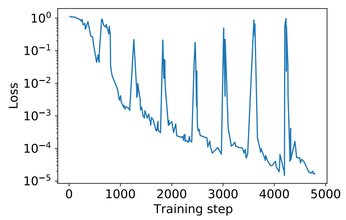

Figure 1. Projection discontinuity problem.

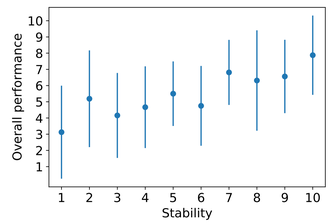

Figure 2. Relation between stability and overall Performance.

Projection Discontinuity Problem

- We plot the training curve of the DiffPruning on the CB task which adjusts the mask every 600 training steps.
- Each time we change the mask, the training error will go back to almost the same value as its initial loss.
- This result shows that changing the mask severely affects the training procedure due to the projection discontinuity problem.

Relation between Stability and Overall Performance

- Emperically verify Theorem 2 which shows that stability implies better generalization.
- Two ranks are positively correlated indicating that stabler models usually have better generalization capability.
- Spearman's rank correlation coefficient (Spearman, 1904) for the two ranks. It can be denoted as $\rho = \frac{\mathrm{cov}(R(S),R(V))}{\sigma_{R(S)}\sigma_{R(V)}}$, $\rho = 0.4356$ with p-value$= 0.000014 < 0.05$.

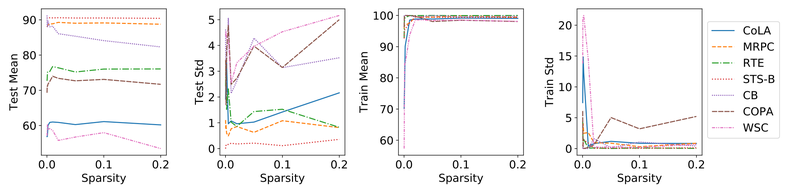

Figure 3. Effectiveness of sparsity.

- Show how the overall performance and the stability change as we change the sparsity in SAM.
- As the sparsity ratio decreases, the models become more stable with better generalization. Verifies with predictions in Theorem 1 and Theorem 2.
- If the sparsity ratio drops below a certain threshold, the models become quite unstable and the performance also sees a sharp drop. Because $\hat{R}(\mathcal{A}, S)$ becomes the dominant term and decreasing the sparsity ratio cannot further lower the bound effectively.

https://github.com/fuzihaofzh/AnalyzeParameterEfficientFinetune