The Expressive Power of Low-Rank Adaptation

Paper Reading - 2025/02/20

Fine-Tuning

- Foundation models (e.g., LLMs, Multi-modal) : large-scale neural networks trained on diverse, extensive datasets
- Generalizable representational frameworks, can be adapted to downstream applications through fine-tuning (e.g., GPT→ ChatGPT)
- Parameters: billions or trillions (e.g., GPT-4 1.76 trillion)
- Fine-tuning (type of transfer learning): Computational Cost

→ Parameter-efficient fine-tuning

Low-Rank Adaptation (LoRA)

- Parameter-efficient fine-tuning method

- Full fine-tuning:

$$\max_{\theta \in \Phi} \hat{L}(f_\theta)$$

Initial Param: θ^{t-1} . Update from (θ^{t-1}) to $(\theta^{t-1} + \Delta\theta)$ over Φ.

- LoRA:

$$\max_{\Delta\theta \in \Theta_{\mathrm{k}}} \hat{L}(f_{\theta^{t-1}+\Delta\theta})$$

Where Θ_{k} is a smaller set, $|\Theta_{\mathrm{k}}| \ll |\Phi|$.
Specifically, $\Theta_{\mathrm{k}} = \{\Delta\theta \in R^{m\times n}: \mathrm{rank}(\Delta\theta) \leq k\}$

Paper

- What is the minimum rank of the LoRA adapters required to adapt a (pre-trained) model f to match the functionality of the target model $\bar{f}$?

Published as a conference paper at ICLR 2024

THE EXPRESSIVE POWER OF LOW-RANK ADAPTATION

Yuchen Zeng
Department of Computer Science
University of Wisconsin-Madison
yzeng58@wisc.edu

Kangwook Lee
Department of Electrical and Computer Engineering
University of Wisconsin-Madison
kangwook.lee@wisc.edu

ABSTRACT

Case 1: Linear Model

- Problem:

Frozen Model $\quad f_0(\boldsymbol{x}) = \boldsymbol{W}_L \cdots \boldsymbol{W}_1 \boldsymbol{x} = \left(\prod_{l=1}^{L} \boldsymbol{W}_l\right) \boldsymbol{x}$

Target Model $\quad \bar{f}(\boldsymbol{x}) = \overline{\boldsymbol{W}} \boldsymbol{x}$

$\overline{\boldsymbol{W}}, \boldsymbol{W}_1, \ldots, \boldsymbol{W}_L \in \mathbb{R}^{D \times D}$

- Find $\Delta \boldsymbol{W}_1, \ldots, \Delta \boldsymbol{W}_L$ s.t. $f = \bar{f}$,

Adapted Model $\quad f(\boldsymbol{x}) = (\boldsymbol{W}_L + \Delta \boldsymbol{W}_L) \cdots (\boldsymbol{W}_1 + \Delta \boldsymbol{W}_1) \boldsymbol{x}$

where $\operatorname{rank}(\Delta \boldsymbol{W}_l) \leq R$ for all $l \in [L]$.

Case 1: Linear Model

Lemma 1. *Define error matrix* $\boldsymbol{E} := \overline{\boldsymbol{W}} - \prod_{l=1}^{L} \boldsymbol{W}_l$*, and denote its rank by* $R_{\boldsymbol{E}} = \mathrm{rank}(\boldsymbol{E})$*. For a given LoRA-rank* $R \in [D]$*, assume that all the weight matrices of the frozen model* $(\boldsymbol{W}_l)_{l=1}^{L}$*, and* $\prod_{l=1}^{L} \boldsymbol{W}_l + \mathrm{LR}_r(\boldsymbol{E})$ *are non-singular for all* $r \leq R(L-1)$*. Then, we have the following:*

$$\min_{\Delta \boldsymbol{W}_l : \mathrm{rank}(\Delta \boldsymbol{W}_l) \leq R} \left\| \prod_{l=1}^{L} (\boldsymbol{W}_l + \Delta \boldsymbol{W}_l) - \overline{\boldsymbol{W}} \right\|_2 = \sigma_{RL+1}(\boldsymbol{E}).$$

Thus, when $R \geq \lceil \frac{R_{\boldsymbol{E}}}{L} \rceil$*, the optimal solution satisfies* $\prod_{l=1}^{L} (\boldsymbol{W}_l + \Delta \boldsymbol{W}_l) = \overline{\boldsymbol{W}}$*, implying* $f = \bar{f}$*.*

→ approximate $R \times L$ ranks of the error E.

Proof:

- Solve constrained optimization:

$$\min_{\Delta \boldsymbol{W}_l : \mathrm{rank}(\Delta \boldsymbol{W}_l) \le R} \left\| \prod_{l=1}^{L} (\boldsymbol{W}_l + \Delta \boldsymbol{W}_l) - \overline{\boldsymbol{W}} \right\|_2 .$$

By subtracting $\prod_{l=1}^{L} \boldsymbol{W}_l$ from both terms, the constrain optimization problem becomes

$$\min_{\Delta \boldsymbol{W}_l : \mathrm{rank}(\Delta \boldsymbol{W}_l) \le R} \left\| \underbrace{\left(\prod_{l=1}^{L} (\boldsymbol{W}_l + \Delta \boldsymbol{W}_l) - \prod_{l=1}^{L} \boldsymbol{W}_l \right)}_{:= \boldsymbol{A}} - \underbrace{\left(\overline{\boldsymbol{W}} - \prod_{l=1}^{L} \boldsymbol{W}_l \right)}_{:= \boldsymbol{E}} \right\|_2 . \quad (2)$$

To perform analysis on (2), we start with the analysis of $\boldsymbol{A}$ as follows:

$$\begin{aligned} \boldsymbol{A} &= \prod_{l=1}^{L} (\Delta \boldsymbol{W}_l + \boldsymbol{W}_l) - \prod_{l=1}^{L} \boldsymbol{W}_l \\ &= \Delta \boldsymbol{W}_L \prod_{l=1}^{L-1} (\Delta \boldsymbol{W}_l + \boldsymbol{W}_l) + \boldsymbol{W}_L \prod_{l=1}^{L-1} (\Delta \boldsymbol{W}_l + \boldsymbol{W}_l) - \prod_{l=1}^{L} \boldsymbol{W}_l . \end{aligned}$$

At this point, it becomes clear that this expression can be iteratively decomposed. Following this pattern, we can express $\boldsymbol{A}$ as:

$$\begin{aligned}\boldsymbol{A} =& \Delta \boldsymbol{W}_L \prod_{l=1}^{L-1} (\Delta \boldsymbol{W}_l + \boldsymbol{W}_l) + \boldsymbol{W}_L \Delta \boldsymbol{W}_{L-1} \prod_{l=1}^{L-2} (\Delta \boldsymbol{W}_l + \boldsymbol{W}_l) \\ &+ \ldots + (\prod_{l=2}^{L} \boldsymbol{W}_l)(\Delta \boldsymbol{W}_1 + \boldsymbol{W}_1) - \prod_{l=1}^{L} \boldsymbol{W}_l \\ =& \sum_{l=1}^{L} \underbrace{\left[(\prod_{i=l+1}^{L} \boldsymbol{W}_i) \Delta \boldsymbol{W}_l (\prod_{i=1}^{l-1} (\boldsymbol{W}_i + \Delta \boldsymbol{W}_i)) \right]}_{:= \boldsymbol{A}_l}.\end{aligned} \tag{3}$$

In this final form, $\boldsymbol{A}$ is decomposed as $\boldsymbol{A} = \sum_{l=1}^{L} \boldsymbol{A}_l$. It is important to note that $\mathrm{rank}(\boldsymbol{A}_l) \leq \mathrm{rank}(\Delta \boldsymbol{W}_l) \leq R$. Consequently, $\mathrm{rank}(\boldsymbol{A}) \leq \sum_{l=1}^{L} \mathrm{rank}(\boldsymbol{A}_l) \leq RL$.

Then, the optimization problem (2) can be relaxed into a low-rank approximation problem

$$(2) \geq \min_{\boldsymbol{A}:\mathrm{rank}(\boldsymbol{A})\leq RL} \left\|\boldsymbol{A}-\boldsymbol{E}\right\|_2 , \tag{4}$$

$$\min_{\Delta \boldsymbol{W}_l:\mathrm{rank}(\Delta \boldsymbol{W}_l)\leq R} \left\| \underbrace{\left(\prod_{l=1}^{L}(\boldsymbol{W}_l + \Delta \boldsymbol{W}_l) - \prod_{l=1}^{L}\boldsymbol{W}_l\right)}_{:=\boldsymbol{A}} - \underbrace{\left(\overline{\boldsymbol{W}} - \prod_{l=1}^{L}\boldsymbol{W}_l\right)}_{:=\boldsymbol{E}} \right\|_2 . \tag{2}$$

where the optimal solution is $\boldsymbol{A} = \mathrm{LR}_{RL\wedge R_{\boldsymbol{E}}}(\boldsymbol{E}) := \boldsymbol{E}'$. Therefore, if we can identify rank-R or lower matrices $(\Delta \boldsymbol{W}_l)_{l=1}^{L}$ such that

$$\underbrace{\prod_{l=1}^{L}(\boldsymbol{W}_l + \Delta \boldsymbol{W}_l) - \prod_{l=1}^{L}\boldsymbol{W}_l}_{:=\boldsymbol{A}} = \underbrace{\mathrm{LR}_{RL\wedge R_{\boldsymbol{E}}}(\overline{\boldsymbol{W}} - \prod_{l=1}^{L}\boldsymbol{W}_l)}_{:=\boldsymbol{E}'}, \tag{5}$$

Denote $R_{\boldsymbol{E}'} = RL \wedge R_{\boldsymbol{E}}$. To derive the explicit form of $\boldsymbol{E}'$, we first refer to the SVD of $\boldsymbol{E}$ as

$$\boldsymbol{E} = \boldsymbol{U}\boldsymbol{D}\boldsymbol{V}^\top,$$

where $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthonormal matrices and the first $R_{\boldsymbol{E}}$ diagonal entries of $\boldsymbol{D}$ are non-zero, with all remaining entries being zero. Based on this, $\boldsymbol{E}'$ is expressed as

$$\boldsymbol{E}' = \boldsymbol{U}\boldsymbol{D}\boldsymbol{I}_{1:RL,D}\boldsymbol{V}^\top.$$

Having already derived the decomposition $\boldsymbol{A} = \sum_{l=1}^{L} \boldsymbol{A}_l$, we next aim to decompose $\boldsymbol{E}'$ as $\boldsymbol{E}' = \sum_{l=1}^{L} \boldsymbol{E}'\boldsymbol{Q}_l$, where $\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_L \in \mathbb{R}^{D \times D}$. The goal now shifts to identifying $\Delta\boldsymbol{W}_l, \boldsymbol{Q}_l$ such that $\boldsymbol{A}_l = \boldsymbol{E}'\boldsymbol{Q}_l$ for each $l \in [L]$. Achieving this would complete the proof of (5).

Therefore, our goal becomes finding $\Delta\boldsymbol{W}_1, \ldots, \Delta\boldsymbol{W}_L$ with $\operatorname{rank}(\Delta\boldsymbol{W}_l) \leq R$ for all $l \in [L]$ such that

$$\boldsymbol{A}_l = \Big(\prod_{i=l+1}^{L} \boldsymbol{W}_i\Big)\Delta\boldsymbol{W}_l\Big(\prod_{i=1}^{l-1}(\boldsymbol{W}_i + \Delta\boldsymbol{W}_i)\Big) = \boldsymbol{E}'\boldsymbol{Q}_l, \quad \text{for all } l \in [L]. \tag{6}$$

One sufficient condition for achieving (6) is that the decomposed matrices $\boldsymbol{Q}_1, \boldsymbol{Q}_L$ and low-rank adapters $\Delta\boldsymbol{W}_1, \ldots, \Delta\boldsymbol{W}_L$ meet the following conditions:

$$\sum_{l=1}^{L} \boldsymbol{E}'\boldsymbol{Q}_l = \boldsymbol{E}', \tag{7}$$

$$\Delta\boldsymbol{W}_l = (\prod_{i=l+1}^{L} \boldsymbol{W}_i)^{-1}\boldsymbol{E}'\boldsymbol{Q}_l(\prod_{i=1}^{l-1}(\boldsymbol{W}_i + \Delta\boldsymbol{W}_i))^{-1}, \text{ for all } l \in [L] \tag{8}$$

$$\operatorname{rank}(\Delta\boldsymbol{W}_l) \le R, \text{ for all } l \in [L], \tag{9}$$

$$\operatorname{rank}(\boldsymbol{W}_l + \Delta\boldsymbol{W}_l) = D, \text{ for all } l \in [L-1]. \tag{10}$$

$$\boldsymbol{A}_l = (\prod_{i=l+1}^{L} \boldsymbol{W}_i)\Delta\boldsymbol{W}_l(\prod_{i=1}^{l-1}(\boldsymbol{W}_i + \Delta\boldsymbol{W}_i)) = \boldsymbol{E}'\boldsymbol{Q}_l, \quad \text{for all } l \in [L].$$

- Consider

$$E = (\sigma_1 u_1 v_1^T + \cdots + \sigma_R u_R v_R^T) + (\ldots.) + (\ldots)$$

We will show that the matrices $(\boldsymbol{Q}_l)_{l=1}^{L}$ defined by

$$\boldsymbol{Q}_l = \boldsymbol{V}\boldsymbol{I}_{(R(l-1)+1)\wedge R_{\boldsymbol{E}'}:Rl\wedge R_{\boldsymbol{E}'},D}\boldsymbol{V}^\top, \quad \text{for all } l \in [L], \tag{11}$$

$$\Delta\boldsymbol{W}_l = (\prod_{i=l+1}^{L}\boldsymbol{W}_i)^{-1}\boldsymbol{E}'\boldsymbol{Q}_l(\prod_{i=1}^{l-1}(\boldsymbol{W}_i + \Delta\boldsymbol{W}_i))^{-1}, \text{ for all } l \in [L]$$

When $l = 1$. We begin by examining the three conditions (8), (9) and (10) under the base case $l = 1$. We first determine $\boldsymbol{Q}_1$ and $\Delta\boldsymbol{W}_1$ based on (11) and (8):

$$\Delta\boldsymbol{W}_1 = (\prod_{i=2}^{L}\boldsymbol{W}_i)^{-1}\boldsymbol{E}'\boldsymbol{Q}_1 \tag{12}$$

By the choice of $\Delta\boldsymbol{W}_1$, we satisfy the condition (8). Moreover, it directly follows that $\text{rank}(\Delta\boldsymbol{W}_1) \leq \text{rank}(\boldsymbol{Q}_1) = R$, thereby fulfilling the rank constraint in (9).

Therefore, we just need to prove that $\boldsymbol{W}_1 + \Delta\boldsymbol{W}_1$ is full-rank, as required by condition (10). To compute $\operatorname{rank}(\boldsymbol{W}_1 + \Delta\boldsymbol{W}_1)$, we proceed as follows:

$$
\begin{aligned}
&\operatorname{rank}(\boldsymbol{W}_1 + \Delta\boldsymbol{W}_1) \\
&\overset{(12)}{=} \operatorname{rank}(\boldsymbol{W}_1 + (\prod_{i=2}^{L}\boldsymbol{W}_i)^{-1}\boldsymbol{E}'\boldsymbol{Q}_1) && \text{(Substituting for } \Delta\boldsymbol{W}_1\text{)} \\
&= \operatorname{rank}((\prod_{i=1}^{L}\boldsymbol{W}_i) + \boldsymbol{E}'\boldsymbol{Q}_1) && \text{(Left multiplying with invertible } (\prod_{i=2}^{L}\boldsymbol{W}_i)^{-1}\text{)} \\
&= \operatorname{rank}((\prod_{i=1}^{L}\boldsymbol{W}_i) + \operatorname{LR}_{R\wedge R_{\boldsymbol{E}'}}(\boldsymbol{E})). && \text{(Simplifying)}
\end{aligned}
$$

Given the assumption that $\prod_{l=1}^{L}\boldsymbol{W}_l + \operatorname{LR}_r(\boldsymbol{E})$ is full rank for all $r \leq R(L-1)$, $\operatorname{rank}(\boldsymbol{W}_1 + \Delta\boldsymbol{W}_1) = \operatorname{rank}((\prod_{i=1}^{L}\boldsymbol{W}_i) + \operatorname{LR}_{R\wedge R_{\boldsymbol{E}'}}(\boldsymbol{E})) = D$, satisfying the last condition (10).

When $l > 1$. Consider $l = 2, \ldots, L$. We assume that for $i \in [l-1]$, we have determined matrices $\boldsymbol{Q}_i$ and $\Delta \boldsymbol{W}_i$ based on (11) and (8), respectively, and we assume that they satisfy the conditions (8), (9), and (10).

First, under the induction assumption that $\boldsymbol{W}_i + \Delta \boldsymbol{W}_i$ is invertible for all $i \in [l-1]$, to achieve $\boldsymbol{A}_l = \boldsymbol{E}'\boldsymbol{Q}_l$, we set $\Delta \boldsymbol{W}_l$ based on (8). This definition ensures $\operatorname{rank}(\Delta \boldsymbol{W}_l) \leq \operatorname{rank}(\boldsymbol{Q}_l) = R$, thereby satisfying the condition (9). To prove that $\boldsymbol{W}_l + \Delta \boldsymbol{W}_l$ is full-rank (condition (10)), we focus on computing $\operatorname{rank}(\boldsymbol{W}_l + \Delta \boldsymbol{W}_l)$. We proceed as follows:

$$
\begin{aligned}
&\operatorname{rank}(\boldsymbol{W}_l + \Delta \boldsymbol{W}_l) \\
&\overset{(8)}{=} \operatorname{rank}\Big(\boldsymbol{W}_l + \Big(\prod_{i=l+1}^{L} \boldsymbol{W}_i\Big)^{-1} \boldsymbol{E}'\boldsymbol{Q}_l\Big(\prod_{i=1}^{l-1} (\boldsymbol{W}_i + \Delta \boldsymbol{W}_i)^{-1}\Big)\Big) && \text{(Substituting for } \Delta \boldsymbol{W}_l\text{)} \\
&= \operatorname{rank}\Big(\boldsymbol{I}_D + \Big(\prod_{i=l}^{L} \boldsymbol{W}_i\Big)^{-1} \boldsymbol{E}'\boldsymbol{Q}_l\Big(\prod_{i=1}^{l-1} (\boldsymbol{W}_i + \Delta \boldsymbol{W}_i)\Big)^{-1}\Big) && \text{(Left multiplying invertible } \boldsymbol{W}_l^{-1}\text{)} \\
&= \operatorname{rank}\Big(\prod_{i=1}^{l-1} (\boldsymbol{W}_i + \Delta \boldsymbol{W}_i) + \Big(\prod_{i=l}^{L} \boldsymbol{W}_i\Big)^{-1} \boldsymbol{E}'\boldsymbol{Q}_l\Big) && \text{(Right multiplying invertible } \prod_{i=1}^{l-1} (\boldsymbol{W}_i + \Delta \boldsymbol{W}_i)\text{)} \\
&= \operatorname{rank}\Big((\boldsymbol{W}_{l-1} + \Delta \boldsymbol{W}_{l-1}) \prod_{i=1}^{l-2} (\boldsymbol{W}_i + \Delta \boldsymbol{W}_i) + \Big(\prod_{i=l}^{L} \boldsymbol{W}_i\Big)^{-1} \boldsymbol{E}'\boldsymbol{Q}_l\Big) && \text{(Rearranging terms)}
\end{aligned}
$$

$$
\begin{aligned}
&\overset{(8)}{=} \operatorname{rank}\Big((\boldsymbol{W}_{l-1} + (\prod_{i=l}^{L} \boldsymbol{W}_i)^{-1} \boldsymbol{E}' \boldsymbol{Q}_{l-1} (\prod_{i=1}^{l-2} (\boldsymbol{W}_i + \Delta \boldsymbol{W}_i))^{-1}) \prod_{i=1}^{l-2} (\boldsymbol{W}_i + \Delta \boldsymbol{W}_i) \\
&\qquad + (\prod_{i=l}^{L} \boldsymbol{W}_i)^{-1} \boldsymbol{E}' \boldsymbol{Q}_l\Big) && \text{(Substituting for } \Delta \boldsymbol{W}_{l-1}) \\
&= \operatorname{rank}\Big((\prod_{i=l-1}^{L} \boldsymbol{W}_i + \boldsymbol{E}' \boldsymbol{Q}_{l-1} (\prod_{i=1}^{l-2} (\boldsymbol{W}_i + \Delta \boldsymbol{W}_i))^{-1}) \prod_{i=1}^{l-2} (\boldsymbol{W}_i + \Delta \boldsymbol{W}_i) \\
&\qquad + \boldsymbol{E}' \boldsymbol{Q}_l\Big) && \text{(Left multiplying } \prod_{i=l}^{L} \boldsymbol{W}_i) \\
&= \operatorname{rank}\left((\prod_{i=l-1}^{L} \boldsymbol{W}_i \prod_{i=1}^{l-2} (\boldsymbol{W}_i + \Delta \boldsymbol{W}_i) + \boldsymbol{E}' \boldsymbol{Q}_{l-1} + \boldsymbol{E}' \boldsymbol{Q}_l\right) && \text{(Rearranging terms)} \\
&= \cdots \\
&= \operatorname{rank}(\prod_{i=1}^{L} \boldsymbol{W}_i + \boldsymbol{E}' (\sum_{i=1}^{l} \boldsymbol{Q}_i)) && \text{(Taking similar steps)} \\
&= \operatorname{rank}(\prod_{i=1}^{L} \boldsymbol{W}_i + \mathrm{LR}_{Rl \wedge R_{\boldsymbol{E}'}}(\boldsymbol{E})). && \text{(Simplifying)}
\end{aligned}
$$

By the assumption that $\prod_{l=1}^{L} \boldsymbol{W}_l + \mathrm{LR}_r(\boldsymbol{E})$ is full-rank for $r \leq R(L-1)$ and consequently, $\operatorname{rank}(\boldsymbol{W}_l + \Delta \boldsymbol{W}_l) = \operatorname{rank}(\prod_{i=1}^{L} \boldsymbol{W}_i + \mathrm{LR}_{Rl \wedge R_{\boldsymbol{E}'}}(\boldsymbol{E})) = D$, satisfying the last condition (10).

$$\min_{\Delta \boldsymbol{W}_l:\mathrm{rank}(\Delta \boldsymbol{W}_l)\leq R} \left\| \underbrace{\left(\prod_{l=1}^{L}(\boldsymbol{W}_l + \Delta \boldsymbol{W}_l) - \prod_{l=1}^{L} \boldsymbol{W}_l\right)}_{:=\boldsymbol{A}} - \underbrace{\left(\overline{\boldsymbol{W}} - \prod_{l=1}^{L} \boldsymbol{W}_l\right)}_{:=\boldsymbol{E}} \right\|_2$$

$$(2) \geq \min_{\boldsymbol{A}:\mathrm{rank}(\boldsymbol{A})\leq RL} \|\boldsymbol{A} - \boldsymbol{E}\|_2$$

$$\underbrace{\prod_{l=1}^{L}(\boldsymbol{W}_l + \Delta \boldsymbol{W}_l) - \prod_{l=1}^{L} \boldsymbol{W}_l}_{:=\boldsymbol{A}} = \underbrace{\mathrm{LR}_{RL\wedge R_{\boldsymbol{E}}}(\overline{\boldsymbol{W}} - \prod_{l=1}^{L} \boldsymbol{W}_l)}_{:=\boldsymbol{E}'},$$

$$\boldsymbol{A} = \textstyle\sum_{l=1}^{L} \boldsymbol{A}_l, \qquad \displaystyle\sum_{l=1}^{L} \boldsymbol{E}'\boldsymbol{Q}_l = \boldsymbol{E}'$$

$$\boldsymbol{A}_l = (\prod_{i=l+1}^{L} \boldsymbol{W}_i)\Delta \boldsymbol{W}_l(\prod_{i=1}^{l-1}(\boldsymbol{W}_i + \Delta \boldsymbol{W}_i)) = \boldsymbol{E}'\boldsymbol{Q}_l,$$

$$\Delta \boldsymbol{W}_l = (\prod_{i=l+1}^{L} \boldsymbol{W}_i)^{-1}\boldsymbol{E}'\boldsymbol{Q}_l(\prod_{i=1}^{l-1}(\boldsymbol{W}_i + \Delta \boldsymbol{W}_i))^{-1}$$

Case 2: 1-Layer ReLU FNN

- Suppose target is 1-layer ReLU FNN

Lemma 9 (Detailed version of Lemma 2). *Define error matrix $\boldsymbol{E} := \overline{\boldsymbol{W}}_1 - \prod_{l=1}^{L} \boldsymbol{W}_l$, with its rank represented by $R_{\boldsymbol{E}} = \operatorname{rank}(\boldsymbol{E})$. Consider a LoRA-rank $R \in [D]$. Assume that the weight matrices $\boldsymbol{W}_1, \ldots, \boldsymbol{W}_L \in \mathbb{R}^{D \times D}$ and $\prod_{l=1}^{L} \boldsymbol{W}_l + \operatorname{LR}_r(\boldsymbol{E})$ for all $r \leq R(L-1)$ are non-singular. Let $\mathbf{x}$ be a random input sampled from a distribution with bounded support $\mathcal{X}$ and let $\Sigma = \mathbb{E}\mathbf{x}\mathbf{x}^\top$. Then, there exists rank-R or lower matrices $\Delta\boldsymbol{W}_1, \ldots, \Delta\boldsymbol{W}_L \in \mathbb{R}^{D \times D}$ and bias vectors $\widehat{\boldsymbol{b}}_1, \ldots, \widehat{\boldsymbol{b}}_L \in \mathbb{R}^D$ such that for any input $\boldsymbol{x} \in \mathcal{X}$,*

$$f(\boldsymbol{x}) - \bar{f}(\boldsymbol{x}) = \mathtt{ReLU}\left(\left(\operatorname{LR}_{RL \wedge R_{\boldsymbol{E}}}(\overline{\boldsymbol{W}}_1 - \prod_{l=1}^{L} \boldsymbol{W}_l) - (\overline{\boldsymbol{W}}_1 - \prod_{l=1}^{L} \boldsymbol{W}_l)\right) \boldsymbol{x}\right).$$

Therefore, when $R \geq \lceil R_{\boldsymbol{E}}/L \rceil$, the adapted model exactly approximates the target model, i.e., $f(\boldsymbol{x}) = \bar{f}(\boldsymbol{x})$ for all $\boldsymbol{x} \in \mathcal{X}$.

Furthermore, let $\mathbf{x}$ be a random input sampled from a distribution with bounded support $\mathcal{X}$ and let $\Sigma = \mathbb{E}\mathbf{x}\mathbf{x}^\top$. Then, the expected squared error is bounded as

$$\mathbb{E}\left\|f(\mathbf{x}) - \bar{f}(\mathbf{x})\right\|_2^2 \leq \|\Sigma\|_F \, \sigma^2_{RL \wedge R_{\boldsymbol{E}}+1}(\overline{\boldsymbol{W}}_1 - \prod_{l=1}^{L} \boldsymbol{W}_l).$$

Proof:

Linearization. The main challenge here stems from the non-linearities introduced by the ReLU activation function. To remove the non-linearities in the first $L-1$ layers of updated model f, since the input space $\mathcal{X}$ is bounded, we can set all the entries of $\widehat{\boldsymbol{b}}_1, \ldots, \widehat{\boldsymbol{b}}_{L-1}$ sufficiently large, thereby

activating all ReLUs in the first $L-1$ layers of f. Consequently, we have

$$
\begin{aligned}
f(\boldsymbol{x}) &= \mathtt{ReLU}((\boldsymbol{W}_L + \Delta \boldsymbol{W}_L)\boldsymbol{z}_{L-1} + \widehat{\boldsymbol{b}}_L) \\
&= \mathtt{ReLU}\left((\boldsymbol{W}_L + \Delta \boldsymbol{W}_L)\mathtt{ReLU}((\boldsymbol{W}_{L-1} + \Delta \boldsymbol{W}_{L-1})\boldsymbol{z}_{L-2} + \widehat{\boldsymbol{b}}_{L-1}) + \widehat{\boldsymbol{b}}_L\right) \\
&= \mathtt{ReLU}\left((\boldsymbol{W}_L + \Delta \boldsymbol{W}_L)((\boldsymbol{W}_{L-1} + \Delta \boldsymbol{W}_{L-1})\boldsymbol{z}_{L-2} + \widehat{\boldsymbol{b}}_{L-1}) + \widehat{\boldsymbol{b}}_L\right) \\
&= \mathtt{ReLU}\left((\boldsymbol{W}_L + \Delta \boldsymbol{W}_L)(\boldsymbol{W}_{L-1} + \Delta \boldsymbol{W}_{L-1})\boldsymbol{z}_{L-2} + (\boldsymbol{W}_L + \Delta \boldsymbol{W}_L)\widehat{\boldsymbol{b}}_{L-1} + \widehat{\boldsymbol{b}}_L\right) \\
&= \cdots \\
&= \mathtt{ReLU}\left(\prod_{l=1}^{L}(\boldsymbol{W}_l + \Delta \boldsymbol{W}_l)\boldsymbol{x} + (\sum_{l=1}^{L-1}\prod_{i=l+1}^{L}(\boldsymbol{W}_i + \Delta \boldsymbol{W}_i)\widehat{\boldsymbol{b}}_l) + \widehat{\boldsymbol{b}}_L\right),
\end{aligned}
$$

which is equivalent to a single-layer ReLU neural network with weight matrix $\prod_{l=1}^{L}(\boldsymbol{W}_l + \Delta \boldsymbol{W}_l)$ and bias vector $(\sum_{l=1}^{L-1}\prod_{i=l+1}^{L}(\boldsymbol{W}_i + \Delta \boldsymbol{W}_i)\widehat{\boldsymbol{b}}_l) + \widehat{\boldsymbol{b}}_L$.

Parameter Alignment. To match the updated model $f(\boldsymbol{x})$ and target model $\bar{f}(\boldsymbol{x})$, we proceed as follows. For weight matrix, Lemma 7 guarantees the existence of rank-R or lower matrices $\Delta\boldsymbol{W}_1, \dots, \Delta\boldsymbol{W}_L \in \mathbb{R}^{D\times D}$ such that

$$\prod_{l=1}^{L}(\boldsymbol{W}_l + \Delta\boldsymbol{W}_l) = \prod_{l=1}^{L}\boldsymbol{W}_l + \mathrm{LR}_{RL\wedge R_{\boldsymbol{E}}}(\overline{\boldsymbol{W}} - \prod_{l=1}^{L}\boldsymbol{W}_l). \tag{14}$$

For the bias vector, we set $\widehat{\boldsymbol{b}}_L = \bar{\boldsymbol{b}}_1 - \sum_{l=1}^{L-1}\prod_{i=l+1}^{L}(\boldsymbol{W}_i + \Delta\boldsymbol{W}_i)\widehat{\boldsymbol{b}}_l$ such that $\sum_{l=1}^{L-1}\prod_{i=l+1}^{L}(\boldsymbol{W}_i + \Delta\boldsymbol{W}_i)\widehat{\boldsymbol{b}}_l + \widehat{\boldsymbol{b}}_L = \bar{\boldsymbol{b}}_1$. Therefore, we obtain

$$f(\boldsymbol{x}) - \bar{f}(\boldsymbol{x}) = \mathtt{ReLU}\left(\left(\mathrm{LR}_{RL\wedge R_{\boldsymbol{E}}}(\overline{\boldsymbol{W}}_1 - \prod_{l=1}^{L}\boldsymbol{W}_l) - (\overline{\boldsymbol{W}}_1 - \prod_{l=1}^{L}\boldsymbol{W}_l)\right)\boldsymbol{x}\right).$$

Error Derivation. We compute the expected squared error as follows:

$$\begin{aligned}
&\mathbb{E}\left\|f(\mathbf{x}) - \bar{f}(\mathbf{x})\right\|_2^2 \\
&\leq \mathbb{E}\left\|\left(\mathrm{LR}_{RL\wedge R_{\boldsymbol{E}}}(\overline{\boldsymbol{W}}_1 - \prod_{l=1}^{L}\boldsymbol{W}_l) - (\overline{\boldsymbol{W}}_1 - \prod_{l=1}^{L}\boldsymbol{W}_l)\right)\mathbf{x}\right\|_2^2 && \text{(ReLU is 1-Lipschitz)} \\
&\overset{(1)}{\leq} \left\|\mathrm{LR}_{RL\wedge R_{\boldsymbol{E}}}(\overline{\boldsymbol{W}}_1 - \prod_{l=1}^{L}\boldsymbol{W}_l) - (\overline{\boldsymbol{W}}_1 - \prod_{l=1}^{L}\boldsymbol{W}_l)\right\|_2^2 \mathbb{E}\left\|\mathbf{x}\right\|_2^2 \\
&= \left\|\Sigma\right\|_{\mathrm{F}} \sigma^2_{RL\wedge R_{\boldsymbol{E}}+1}(\overline{\boldsymbol{W}}_1 - \prod_{l=1}^{L}\boldsymbol{W}_l). && \text{(By the definition of } \mathrm{LR}_{RL\wedge R_{\boldsymbol{E}}}(\cdot))
\end{aligned}$$

This completes the proof. $\square$

Case 3: L-Layer ReLU FNN

- Model Partition:

Example 1. *Consider the case where $\bar{L} = 2$ and $L = 4$. We view a two-layer target model $\bar{f}$ as a composition of two one-layer ReLU FNNs. Accordingly, we partition the four-layer adapted model f into two submodels, each consisting of two layers. For each layer in the target model, we utilize two corresponding layers in the frozen/adapted model for approximation. This problem then simplifies into a one-layer FNN approximation problem, which has already been addressed in Lemma 2.*

Based on this example, we introduce a ordered partition $\mathcal{P} = \{P_1, \ldots, P_{\bar{L}}\}$ to partition the layers in the adapted model f, where $\bigcup_{i=1}^{\bar{L}} P_i = [L]$. Each element $P_i \in \mathcal{P}$ consists of consecutive integers. Given a partition $\mathcal{P}$, each element P_i specifies that the layers with index $l \in P_i$ in the adapted model will be used to approximate the i-th layer in the target model. Example 1, which uses every two layers in the adapted model to approximate each layer in the target model, can be considered as a partition represented as $\{\{1,2\},\{3,4\}\}$. Similarly, we extend this simple uniform partition into general cases for $\bar{L}$-layer target FNN and L-layer frozen FNN:

$$\mathcal{P}^{\mathsf{u}} = \left\{P_1^{\mathsf{u}}, \ldots, P_{\bar{L}}^{\mathsf{u}}\right\} := \left\{\{1, \ldots, M\}, \{M+1, \ldots, 2M\}, \ldots, \{(\bar{L}-1)M+1, \ldots, L\}\right\},$$

where $M := \lfloor L/\bar{L} \rfloor$. The uniform partition indicates that every M layers in the adapted model are employed to approximate each layer in the target model. We use $\prod_{l \in P_i} \boldsymbol{W}_l$ to denote the product of the weight matrices from the layers $l \in P_i$, with the later layer positioned to the left and the earlier layer to the right in the matrix product. For example, $\prod_{l \in P_1^{\mathsf{u}}} \boldsymbol{W}_l = \prod_{l=1}^{M} \boldsymbol{W}_l = \boldsymbol{W}_M \cdots \boldsymbol{W}_1$.

Case 3: L-Layer ReLU FNN

Theorem 5. *Define the approximation error of i-th layer as* $E_i = \sigma_{RM+1}(\overline{\boldsymbol{W}}_i - \prod_{l \in P_i^u} \boldsymbol{W}_l)$, *and the magnitude of the parameters and the input as* $\beta := \max_{i \in [\overline{L}]} \left(\sqrt{\|\Sigma\|_F} \prod_{j=1}^{i} \|\overline{\boldsymbol{W}}_j\|_F + \sum_{j=1}^{i} \prod_{k=j+1}^{i-1} \|\overline{\boldsymbol{W}}_k\|_F \|\overline{\boldsymbol{b}}_j\|_2 \right) \bigvee \sqrt{\|\Sigma\|_F}$.

Under Assumption 1, there exists rank-R or lower matrices $(\Delta \boldsymbol{W}_l)_{l=1}^{L}$ with $\Delta \boldsymbol{W}_l \in \mathbb{R}^{D \times D}$ and bias vectors $(\widehat{\boldsymbol{b}}_l)_{l=1}^{L}$ with $\widehat{\boldsymbol{b}}_l \in \mathbb{R}^D$ such that for input $\mathbf{x} \in \mathcal{X}$ with $\mathbb{E}\mathbf{x}\mathbf{x}^\top = \Sigma$,

$$\mathbb{E} \left\| f(\mathbf{x}) - \overline{f}(\mathbf{x}) \right\|_2 \leq \beta \sum_{i=1}^{\overline{L}} \max_{k \in [\overline{L}]} \left(\|\overline{\boldsymbol{W}}_k\|_F + E_k \right)^{\overline{L}-i} E_i.$$

Proof:

Model Decomposition. We partition the adapted model f into $\overline{L}$ sub-models, each defined as

$$f_i(\cdot) = \texttt{FNN}_{\overline{L},D}(\cdot; (\boldsymbol{W}_l + \Delta\boldsymbol{W}_l)_{l\in P_i^{\mathrm{u}}}, (\widehat{\boldsymbol{b}}_l)_{l\in P_i^{\mathrm{u}}}), \quad i \in [\overline{L}].$$

In a similar manner, we break down $\overline{f}$ into $\overline{L}$ sub-models, each is a one-layer FNN:

$$\overline{f}_i(\cdot) = \texttt{FNN}_{1,D}(\cdot; \overline{\boldsymbol{W}}_i, \overline{\boldsymbol{b}}_i),\ i \in [\overline{L}].$$

We can then express $f(\boldsymbol{x})$ and $\overline{f}(\boldsymbol{x})$ as compositions of their respective sub-models:

$$f(\cdot) = f_{\overline{L}} \circ \cdots f_1(\cdot), \quad \overline{f}(\cdot) = \overline{f}_{\overline{L}} \circ \cdots \overline{f}_1(\cdot).$$

To analyze the error $\mathbb{E}\left\|f(\mathbf{x}) - \overline{f}(\mathbf{x})\right\|_2 = \mathbb{E}\left\|f(\mathbf{x}) - \overline{f}(\mathbf{x})\right\|_2$, we consider the error caused by each submodel. Let $\widetilde{R}_i = \operatorname{rank}(\overline{\boldsymbol{W}}_i - \prod_{l\in P_i^{\mathrm{u}}} \boldsymbol{W}_l)$ denote the rank of the discrepancy between the target weight matrix and the frozen weight matrices, where $i \in [\overline{L}]$. By Lemma 9, we can select $\Delta\boldsymbol{W}_1, \ldots, \Delta\boldsymbol{W}_L, \widehat{\boldsymbol{b}}_1, \ldots, \widehat{\boldsymbol{b}}_L$ such that

$$f_i(\boldsymbol{z}) - \overline{f}_i(\boldsymbol{z}) = \texttt{ReLU}\left(\left(\mathrm{LR}_{RL\wedge\widetilde{R}_i}(\overline{\boldsymbol{W}}_i - \prod_{l\in P_i^{\mathrm{u}}} \boldsymbol{W}_l) - (\overline{\boldsymbol{W}}_i - \prod_{l\in P_i^{\mathrm{u}}} \boldsymbol{W}_l)\right)\boldsymbol{z}\right), \tag{15}$$

$$\mathbb{E}\left\|f_i(\mathbf{z}) - \overline{f}_i(\mathbf{z})\right\|_2^2 \leq \left\|\mathbb{E}\mathbf{z}\mathbf{z}^\top\right\|_{\mathrm{F}} \sigma^2_{RL\wedge\widetilde{R}_i+1}(\overline{\boldsymbol{W}}_i - \prod_{l=1}^{L} \boldsymbol{W}_l). \tag{16}$$

Error Decomposition. For submodel $i = 2, \ldots, \bar{L}$, we calculate the expected error of the composition of the first i sub-models,

$$\mathbb{E}\left\|\widehat{\mathbf{z}}_i - \bar{\mathbf{z}}_i\right\|_2 = \mathbb{E}\left\|f_i(\widehat{\mathbf{z}}_{i-1}) - \bar{f}_i(\bar{\mathbf{z}}_{i-1})\right\|_2 \tag{17}$$

$$= \mathbb{E}\left\|(f_i(\widehat{\mathbf{z}}_{i-1}) - f_i(\bar{\mathbf{z}}_{i-1})) + \left(f_i(\bar{\mathbf{z}}_{i-1}) - \bar{f}_i(\bar{\mathbf{z}}_{i-1})\right)\right\|_2 \qquad \text{(Rearranging terms)}$$

$$\leq \underbrace{\mathbb{E}\left\|f_i(\widehat{\mathbf{z}}_{i-1}) - f_i(\bar{\mathbf{z}}_{i-1})\right\|_2}_{A_i} + \underbrace{\mathbb{E}\left\|f_i(\bar{\mathbf{z}}_{i-1}) - \bar{f}_i(\bar{\mathbf{z}}_{i-1})\right\|_2}_{B_i}. \qquad \text{(Applying triangle inequality)}$$

Here A_i represents the error resulting from the discrepancy between the first $i-1$ submodels, while B_i represents the error arising from the mismatch between the i-th submodel.

Computing A_i. We start by computing the error introduced by the first $i-1$ submodels, denoted by A_i:

$$A_i = \mathbb{E}\left\|f_i(\widehat{\mathbf{z}}_{i-1}) - f_i(\bar{\mathbf{z}}_{i-1})\right\|_2 = \mathbb{E}\left\|\mathtt{ReLU}(\widehat{\boldsymbol{W}}_i(\widehat{\mathbf{z}}_{i-1} - \bar{\mathbf{z}}_{i-1}))\right\|_2$$

$$\leq \mathbb{E}\left\|\widehat{\boldsymbol{W}}_i(\widehat{\mathbf{z}}_{i-1} - \bar{\mathbf{z}}_{i-1})\right\|_2 \qquad \text{(ReLU is 1-Lipschitz)}$$

$$\overset{(1)}{\leq} \left\|\widehat{\boldsymbol{W}}_i\right\|_{\mathrm{F}} \mathbb{E}\left\|\widehat{\mathbf{z}}_{i-1} - \bar{\mathbf{z}}_{i-1}\right\|_2. \tag{18}$$

$$\widehat{\boldsymbol{W}}_i = \mathrm{LR}_{RL\wedge\widetilde{R}_i}(\overline{\boldsymbol{W}}_1 - \textstyle\prod_{l\in P_i^{\mathrm{u}}}\boldsymbol{W}_l) + \textstyle\prod_{l\in P_i^{\mathrm{u}}}\boldsymbol{W}_l.$$

$$\prod_{l=1}^{L}(\boldsymbol{W}_l + \Delta\boldsymbol{W}_l) = \prod_{l=1}^{L}\boldsymbol{W}_l + \mathrm{LR}_{RL\wedge R_{\boldsymbol{E}}}(\overline{\boldsymbol{W}} - \prod_{l=1}^{L}\boldsymbol{W}_l).$$

Here,

$$
\begin{aligned}
\left\|\widehat{\boldsymbol{W}}_i\right\|_{\mathrm{F}} &= \left\|\prod_{l\in P_i^{\mathrm{u}}} \boldsymbol{W}_l + \mathrm{LR}_{RM\wedge\widetilde{R}_i}(\overline{\boldsymbol{W}}_i - \prod_{l\in P_i^{\mathrm{u}}} \boldsymbol{W}_l)\right\|_{\mathrm{F}} \\
&= \left\|\overline{\boldsymbol{W}}_i + \left(\prod_{l\in P_i^{\mathrm{u}}} \boldsymbol{W}_l - \overline{\boldsymbol{W}}_i\right) + \mathrm{LR}_{RM\wedge\widetilde{R}_i}(\overline{\boldsymbol{W}}_i - \prod_{l\in P_i^{\mathrm{u}}} \boldsymbol{W}_l)\right\|_{\mathrm{F}} \quad \text{(Rearranging terms)} \\
&\le \|\overline{\boldsymbol{W}}_i\|_{\mathrm{F}} + \left\|\left(\prod_{l\in P_i^{\mathrm{u}}} \boldsymbol{W}_l - \overline{\boldsymbol{W}}_i\right) + \mathrm{LR}_{RM\wedge\widetilde{R}_i}(\overline{\boldsymbol{W}}_i - \prod_{l\in P_i^{\mathrm{u}}} \boldsymbol{W}_l)\right\|_{\mathrm{F}} \\
&\qquad \text{(Applying triangle inequality)} \\
&= \|\overline{\boldsymbol{W}}_i\|_{\mathrm{F}} + \sqrt{\sum_{j=RM\wedge\widetilde{R}_i+1}^{D} \sigma_j^2(\overline{\boldsymbol{W}}_i - \prod_{l\in P_i^{\mathrm{u}}} \boldsymbol{W}_l)} \qquad (19) \\
&\qquad \text{(By the definition of } \overline{\boldsymbol{W}}_i \text{ and } \mathrm{LR}_{RM\wedge\widetilde{R}_i+1}(\cdot)) \\
&\le \max_{k\in[\overline{L}]}(\left\|\overline{\boldsymbol{W}}_k\right\|_{\mathrm{F}} + E_i) \coloneqq \alpha.
\end{aligned}
$$

By combining (18) and (19), we get

$$
A_i \le \max_{k\in[\overline{L}]}\left(\left\|\overline{\boldsymbol{W}}_k\right\|_{\mathrm{F}} + E_i\right) \mathbb{E}\left\|\widehat{\mathbf{z}}_{i-1} - \overline{\mathbf{z}}_{i-1}\right\|_2 \le \alpha \mathbb{E}\left\|\widehat{\mathbf{z}}_{i-1} - \overline{\mathbf{z}}_{i-1}\right\|_2 . \tag{20}
$$

$$
E_i = \sigma_{RM+1}(\overline{\boldsymbol{W}}_i - \textstyle\prod_{l\in P_i^{\mathrm{u}}} \boldsymbol{W}_l)
$$

Computing B_i. We proceed to compute the error associated with the i-th submodel, which we denote as B_i. It can be evaluated as follows:

$$
\begin{aligned}
B_i &= \mathbb{E}\left\|f_i(\bar{\mathbf{z}}_{i-1}) - \bar{f}_i(\bar{\mathbf{z}}_{i-1})\right\|_2 \\
&\overset{(15)}{=} \mathbb{E}\left\|\mathtt{ReLU}\left(\left(\mathrm{LR}_{RM\wedge\widetilde{R}_i}(\overline{\boldsymbol{W}}_i - \prod_{l\in P_i^{\mathrm{u}}}\boldsymbol{W}_l) - (\overline{\boldsymbol{W}}_i - \prod_{l\in P_i^{\mathrm{u}}}\boldsymbol{W}_l)\right)\bar{\mathbf{z}}_{i-1}\right)\right\|_2 \\
&\leq \mathbb{E}\left\|\left(\mathrm{LR}_{RM\wedge\widetilde{R}_i}(\overline{\boldsymbol{W}}_i - \prod_{l\in P_i^{\mathrm{u}}}\boldsymbol{W}_l) - (\overline{\boldsymbol{W}}_i - \prod_{l\in P_i^{\mathrm{u}}}\boldsymbol{W}_l)\right)\bar{\mathbf{z}}_{i-1}\right\|_2 \qquad \text{(ReLU is 1-Lipschitz)} \\
&\overset{(1)}{\leq} \left\|\mathrm{LR}_{RM\wedge\widetilde{R}_i}(\overline{\boldsymbol{W}}_i - \prod_{l\in P_i^{\mathrm{u}}}\boldsymbol{W}_l) - (\overline{\boldsymbol{W}}_i - \prod_{l\in P_i^{\mathrm{u}}}\boldsymbol{W}_l)\right\|_2 \mathbb{E}\left\|\bar{\mathbf{z}}_{i-1}\right\|_2 \\
&= \sigma_{RM\wedge\widetilde{R}_i+1}(\overline{\boldsymbol{W}}_i - \prod_{l\in P_i^{\mathrm{u}}}\boldsymbol{W}_l)\mathbb{E}\left\|\bar{\mathbf{z}}_{i-1}\right\|_2.
\end{aligned}
$$

We can further simplify $\mathbb{E}\left\|\bar{\mathbf{z}}_{i-1}\right\|_2$ as :

$$
\begin{aligned}
&\mathbb{E}\left\|\bar{\mathbf{z}}_{i-1}\right\|_2 \\
&=\mathbb{E}\left\|\mathtt{ReLU}(\overline{\boldsymbol{W}}_{i-1}\bar{\mathbf{z}}_{i-2}+\bar{\boldsymbol{b}}_{i-1})\right\|_2 \\
&=\mathbb{E}\left\|\overline{\boldsymbol{W}}_{i-1}\bar{\mathbf{z}}_{i-2}+\bar{\boldsymbol{b}}_{i-1}\right\|_2 && \text{(ReLU is 1-Lipschitz)} \\
&\leq\left\|\overline{\boldsymbol{W}}_{i-1}\right\|_{\mathrm{F}}\mathbb{E}\left\|\bar{\mathbf{z}}_{i-2}\right\|_2+\left\|\bar{\boldsymbol{b}}_{i-1}\right\|_2 && \text{(Applying triangle inequality and (1))} \\
&\leq\left\|\overline{\boldsymbol{W}}_{i-1}\right\|_{\mathrm{F}}\left(\left\|\overline{\boldsymbol{W}}_{i-2}\right\|_{\mathrm{F}}\mathbb{E}\left\|\bar{\mathbf{z}}_{i-3}\right\|_2+\left\|\bar{\boldsymbol{b}}_{i-2}\right\|_2\right)+\left\|\bar{\boldsymbol{b}}_{i-1}\right\|_2 && \text{(Following the same steps)} \\
&\leq\prod_{j=1}^{i-1}\left\|\overline{\boldsymbol{W}}_{j}\right\|_{\mathrm{F}}\mathbb{E}\left\|\mathbf{x}\right\|_2+\sum_{j=1}^{i-1}\prod_{k=j+1}^{i-1}\left\|\overline{\boldsymbol{W}}_{k}\right\|_{\mathrm{F}}\left\|\bar{\boldsymbol{b}}_{j}\right\|_2 && \text{(Repeating the same steps)} \\
&=\sqrt{\left\|\Sigma\right\|_{\mathrm{F}}}\prod_{j=1}^{i-1}\left\|\overline{\boldsymbol{W}}_{j}\right\|_{\mathrm{F}}+\sum_{j=1}^{i-1}\prod_{k=j+1}^{i-1}\left\|\overline{\boldsymbol{W}}_{k}\right\|_{\mathrm{F}}\left\|\bar{\boldsymbol{b}}_{j}\right\|_2\leq\beta.
\end{aligned}
$$

Therefore, we obtain

$$
B_i\leq\beta\sigma_{RM\wedge\widetilde{R}_i+1}(\overline{\boldsymbol{W}}_i-\prod_{l\in P_i^{\mathrm{u}}}\boldsymbol{W}_l).
$$

Error Composition. Having established upper bounds for A_i and B_i, we next evaluate the expected error for the composition of the first i adapted submodels.

$$\mathbb{E}\left\|\widehat{\mathbf{z}}_i-\bar{\mathbf{z}}_i\right\|_2 \overset{(17)}{\leq} A_i+B_i \overset{(20)}{\leq} \alpha\mathbb{E}\left\|\widehat{\mathbf{z}}_{i-1}-\bar{\mathbf{z}}_{i-1}\right\|_2+B_i \leq \alpha(\alpha\mathbb{E}\left\|\widehat{\mathbf{z}}_{i-2}-\bar{\mathbf{z}}_{i-2}\right\|_2+B_{i-1})+B_i$$
$$=\alpha^2\mathbb{E}\left\|\widehat{\mathbf{z}}_{i-2}-\bar{\mathbf{z}}_{i-2}\right\|_2+\alpha B_{i-1}+B_i \leq \cdots \leq \alpha^{i-1}\mathbb{E}\left\|\widehat{\mathbf{z}}_1-\bar{\mathbf{z}}_1\right\|_2+\sum_{k=2}^{i}\alpha^{i-k}B_k. \tag{21}$$

To compute the overall approximation error of f, which is the composite of all submodels, we have

$$\begin{aligned}
&\mathbb{E}\left\|f(\mathbf{x})-\bar{f}(\mathbf{x})\right\|_2=\mathbb{E}\left\|f(\mathbf{x})-\bar{f}(\mathbf{x})\right\|_2=\mathbb{E}\left\|\widehat{\mathbf{z}}_{\bar{L}}-\bar{\mathbf{z}}_{\bar{L}}\right\|_2\\
&\overset{(21)}{\leq} \alpha^{\bar{L}-1}\mathbb{E}\left\|\widehat{\mathbf{z}}_1-\bar{\mathbf{z}}_1\right\|_2+\sum_{i=2}^{\bar{L}}\alpha^{\bar{L}-i}B_i\\
&\overset{(16)}{\leq} \alpha^{\bar{L}-1}\beta\sigma_{RM\wedge\widetilde{R}_i+1}(\overline{\boldsymbol{W}}_i-\prod_{l\in P_i^{\mathrm{u}}}\boldsymbol{W}_l)+\beta\sum_{i=2}^{\bar{L}}\alpha^{\bar{L}-i}\sigma_{RM\wedge\widetilde{R}_i+1}(\overline{\boldsymbol{W}}_i-\prod_{l\in P_i^{\mathrm{u}}}\boldsymbol{W}_l)\\
&=\beta\sum_{i=1}^{\bar{L}}\alpha^{\bar{L}-i}\sigma_{RM\wedge\widetilde{R}_i+1}(\overline{\boldsymbol{W}}_i-\prod_{l\in P_i^{\mathrm{u}}}\boldsymbol{W}_l)\\
&=\beta\sum_{i=1}^{\bar{L}}\alpha^{\bar{L}-i}\sigma_{RM+1}(\overline{\boldsymbol{W}}_i-\prod_{l\in P_i^{\mathrm{u}}}\boldsymbol{W}_l).
\end{aligned}$$

Substituting α with $\max_{k\in[\bar{L}]}(\left\|\overline{\boldsymbol{W}}_k\right\|_{\mathrm{F}}+E_i)$ concludes the proof. □

Transfer Learning:

Given:

- $\{X_i^{(0)}, Y_i^{(0)}\}_N \sim P^{(0)}$
- $\{X_i^{(1)}, Y_i^{(1)}\}_n \sim P^{(1)}, \mathrm{n} \ll N$

Suppose:

- $y_i^{(0)} = M_L\sigma\left(M_{L-1}\sigma\left(M_{L-2}\sigma\left(\dots.M_2\sigma\left(M_1x_i^{(0)}\right)\right)\right)\right) + \epsilon_i$
- $y_i^{(1)} = M_L\sigma\left(M_{L-1}\sigma\left(M_{L-2}\sigma\left(\dots.M_2\sigma\left((M_1+Z)x_i^{(1)}\right)\right)\right)\right) + \epsilon_i$

Where $rank(Z) \leq k$

- Suppose have exactly $\{\mathrm{M_l}\}_{1:L}$, without estimation error $(\mathrm{n} \ll N)$.
- $\mathrm{W_l} = M_1 + Z$, we can estimate without gradient method to obtain $\mathrm{RW_l}$ for some orthogonal R, and hence the V in $M_1 + Z = UV^T, V^T V = I$

Under the case with Gaussian input $\boldsymbol{x} \in \mathcal{N}(0, \boldsymbol{\Sigma})$, the first and second order score reduce to

$$
\begin{aligned}
S(\boldsymbol{x}) &= \boldsymbol{\Sigma}^{-1}\boldsymbol{x}, \\
T(\boldsymbol{x}) &= \boldsymbol{\Sigma}^{-1}\boldsymbol{x}\boldsymbol{x}^\top\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1}.
\end{aligned}
$$

If we further let $\boldsymbol{z} = \boldsymbol{W}_1\boldsymbol{x}$ and assume that both $\mathbb{E}[yT(\boldsymbol{x})]$ and $\mathbb{E}\left[\nabla_{\boldsymbol{z}}^2 f(\boldsymbol{W}_1\boldsymbol{x})\right]$ are well-defined, then a second-order Stein's formula suggests that

$$
\mathbb{E}[yT(\boldsymbol{x})] = \boldsymbol{W}_1^\top \mathbb{E}\left[\nabla_{\boldsymbol{z}}^2 f(\boldsymbol{W}_1\boldsymbol{x})\right] \cdot \boldsymbol{W}_1. \tag{10}
$$

The equation (10) serves as the basis for estimating $\mathcal{S}_0$. Let $\mathbf{A} = \mathbb{E}[yT(\boldsymbol{x})]$ and denote its eigenvalue decomposition as $\mathbf{A} = \boldsymbol{W}^\top\boldsymbol{D}\boldsymbol{W}$. The second-order Stein's formula suggests that there exists an orthogonal matrix $\boldsymbol{R} \in \mathbb{R}^{k_1 \times k_1}$ such that $\boldsymbol{W}_1 = \boldsymbol{R}\boldsymbol{W}$. In other words, $\boldsymbol{W}_1$ has the same row space as $\boldsymbol{W}$, regardless of the specific form of link function f. Consequently, $\mathcal{S}_0$ can be obtained from the eigenvalue decomposition of $\boldsymbol{A}$.

$$M_1 + Z = UV^T$$
$$(M_1 + Z)VV^T = UV^TVV^T = UV^T = M_1 + Z$$
$$\mathrm{M}_1(VV^T - I) + Z(VV^T - I) = 0$$

To estimate Z, consider the constrained optimization:

$$\min_{Z:rank(Z)\leq k} |\mathrm{M}_1(VV^T - I) + Z(VV^T - I)|_F^2$$

$$= \min_{Z:rank(Z)\leq k} |Y - ZX|_F^2$$

Where $Y = \mathrm{M}_1(VV^T - I), \mathrm{X} = (I - VV^T)$. There is closed-form solution to this.

- (1) Estimate Z without gradient method
- (2) Parameter-efficient, $\mathrm{rank}(Z) \leq k$, good when $n \ll N$.

- Some potential Scenarios?

$\{X_i^{(0)}, Y_i^{(0)}\}_N \sim P^{(0)}$: Population

$\{X_i^{(1)}, Y_i^{(1)}\}_n \sim P^{(1)}$: Subgroup, $n \ll N$

→low-rank adjustment is enough

To formalize the idea:

(1): Error and noise: $\widehat{M}_1 \neq M_1$, etc

(2): Deal with non-linearity to extend to $M_2 + Z_2, M_3 + Z_3$, ...

$$y_i^{(1)} = (M_L + Z_L)\,\sigma\left((M_{L-1} + Z_{L-1})\sigma\left((M_{L-2} + Z_{L-2})\sigma\left(\dots.(M_2 + Z_2)\sigma\left((M_1 + Z_1)x_i^{(1)}\right)\right)\right)\right) + \epsilon_i$$

$$= (M_L + Z_L)\,\sigma\left((M_{L-1} + Z_{L-1})\sigma((M_{L-2} + Z_{L-2})\sigma(\dots.(M_2 + Z_2)g_1))\right) + \epsilon_i$$

To continue, we would need to deal with truncated normal:

$$g_1 = \sigma\left((M_1 + Z_1)x_i^{(1)}\right)$$

We need the second-order score:

Under the case with Gaussian input $\boldsymbol{x} \in \mathcal{N}(0, \boldsymbol{\Sigma})$, the first and second order score reduce to

$$\begin{aligned} S(\boldsymbol{x}) &= \boldsymbol{\Sigma}^{-1}\boldsymbol{x}, \\ T(\boldsymbol{x}) &= \boldsymbol{\Sigma}^{-1}\boldsymbol{x}\boldsymbol{x}^\top\boldsymbol{\Sigma}^{-1} - \boldsymbol{\Sigma}^{-1}. \end{aligned}$$

If we further let $\boldsymbol{z} = \boldsymbol{W}_1\boldsymbol{x}$ and assume that both $\mathbb{E}[yT(\boldsymbol{x})]$ and $\mathbb{E}\left[\nabla_{\boldsymbol{z}}^2 f(\boldsymbol{W}_1\boldsymbol{x})\right]$ are well-defined, then a second-order Stein's formula suggests that

$$\mathbb{E}[yT(\boldsymbol{x})] = \boldsymbol{W}_1^\top\mathbb{E}\left[\nabla_{\boldsymbol{z}}^2 f(\boldsymbol{W}_1\boldsymbol{x})\right] \cdot \boldsymbol{W}_1. \tag{10}$$

- Randomly initializes $\{\mathrm{M_l}\}_{1:L}$ (they are known), and obtain $\{\mathrm{Z_l}\}_{1:L}$, if $\mathrm{rank}(\mathrm{Z_l})$ is large enough, this can approximate any L-layer FNN

$$y_i^{(1)} = (M_L + Z_L)\,\sigma\left((M_{L-1} + Z_{L-1})\sigma\left((M_{L-2} + Z_{L-2})\sigma\left(\ldots.(M_2 + Z_2)\sigma\left((M_1 + Z_1)x_i^{(1)}\right)\right)\right)\right) + \epsilon_i$$