Less-forgetting Multi-lingual Fine-tuning (Appendix)

In this Appendix, the proofs for Theorem 1, Theorem 2 and Theorem 3 are given in Section 1. Besides, the details of the datasets and computing platform are introduced in Section 2.

1 Proofs

Theorem 1. *Assume $L_p(\theta_f)$ can be approximate by its second order Taylor expansion and θ_p is a minima w.r.t the pre-training loss. Then, we have*

$$F_p \leq \frac{\lambda_p \eta}{2} \| \sum_{k=1}^{K} \sum_{t=1}^{T} w_t^k \nabla L_t(\theta_{k-1}) \|^2. \tag{1}$$

where λ_p is the maximum eigenvalue of $\nabla^2 L_p(\theta_p)$, and K is the number of iterations of fine-tuning.

Proof. For $L_p(\theta_f)$ can be approximate by its second order Taylor expansion, we have

$$L_p(\theta_f) = L_p(\theta_p) + (\theta_f - \theta_p)^\top \nabla L_p(\theta_p) + \frac{1}{2}(\theta_f - \theta_p)^\top \nabla^2 L_p(\theta_p)(\theta_f - \theta_p). \tag{2}$$

Then, the second assumption, θ_p is a minima, presents that $\|\nabla L_p(\theta_p)\| = 0$. Thus,

$$L_p(\theta_f) = L_p(\theta_p) + \frac{1}{2}(\theta_f - \theta_p)^\top \nabla^2 L_p(\theta_p)(\theta_f - \theta_p). \tag{3}$$

Moreover, the minima assumption leads to the conclusion that $\nabla^2 L_p(\theta_p)$ is positive semi-definite; thus, we have

$$L_p(\theta_f) \leq L_p(\theta_p) + \frac{\lambda_p}{2} \|\theta_p - \theta_f\|^2. \tag{4}$$

In multi-lingual fine-tuning,

$$\theta_p - \theta_f = \eta \sum_{k=1}^{K} \sum_{t=1}^{T} w_t^k \nabla L_t(\theta_{k-1}) \tag{5}$$

Overall,

$$F_p = L_p(\theta_f) - L_p(\theta_p) \leq \frac{\lambda_p \eta}{2} \| \sum_{k=1}^{K} \sum_{t=1}^{T} w_t^k \nabla L_t(\theta_{k-1}) \|^2. \tag{6}$$

□

Theorem 2. *Let $\mathcal{H}$ be a Hilbert space of finite dimension N. Let $L_t(\theta_k)$ ($1 \leq t \leq T \leq N$) be T smooth functions of the vector $\theta_k \in \mathcal{H}$, and θ_k^0 a particular admissible design-point. Let w_k^* be the solution of Problem 2 and descent direction $\nabla L = \sum_{t=1}^{T} (w_k^*)^t \nabla L_t(\theta_k)$. Then:*

(i) either $\nabla L = \varnothing$, and $[L_1(\theta_k^0), ..., L_T(\theta_k^0)]^\top$ are pareto stationary at θ_k^0;

(ii) or $\nabla L \neq \varnothing$ and $-\nabla L$ is a descent direction common to all $\{L_t(\theta_k)\}_{t=1}^T$;

Proof. For the first claim, $\nabla L = \varnothing$ means that there exists at least one objective, on which

$$\sum_{t=1}^{T} (w_k^*)^t \nabla L_t(\theta_k^0))^\top L_t(\theta_k^0) < 0. \tag{7}$$

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

Then, obviously, there exists at least one w_k that makes $\nabla L = 0$, which means that θ_k^0 is Pareto stationary.

Furthermore, the second claim is straightforward. Because, when $\nabla L \neq \varnothing$, we can find $(w_k^*)^t$ which makes

$$-\nabla L^\top L_t(\theta_k^0) \leq 0. \tag{8}$$

for every objective. It demonstrates that ∇L is a descent direction common to all $\{L_t(\theta_k^0)\}_{t=1}^T$. □

Theorem 3. *LF-MLF can stop after a finite number of iterations if a Pareto stationary point is reached. Otherwise, If the sequence of iterates $\{\theta_k\}_{k=1}^K$ of the LF-MLF is infinite, it admits a weakly convergent subsequence.*

Proof. In the LF-MLF algorithm, the algorithm will stop if the feasible set is empty. According to Theorem 2, when the feasible set is empty, the model has achieved a Pareto stationary point. Thus, LF-MLF can stop after a finite number of iterations if a Pareto stationary point is reached.

As to the infinite case, since the sequence of values of $\{L_t(\theta_k)\}_{k=1}^K$, is positive and monotone-decreasing, it is bounded. Furthermore, $L_t(\theta_k)$ is infinite whenever θ_k is infinite. In fine-tuning, $\{L_t(\theta_k)\}_{k=1}^K$ are bounded. Therefore, sequence of iterates $\{\theta_k\}_{k=1}^K$ admits a weakly convergent subsequence. □

2 The Details of the Datasets and Computing Platform

For NER, the Wikiann [1] is used, and select 48 languages, i.e., ar, he, vi, id, jv, ms, tl, eu, ml, ta, te, af, nl, en, de, el, bn, hi, mr, ur, fa, fr, it, pt, es, bg, ru, ja, ka, ko, th, sw, yo, my, zh, kk, tr, et, fi, hu, qu, pl, uk, az, lt, pa, gu, ro.

For QA, we use the gold passage version of the Typologically Diverse Question Answering (TyDiQA-GoldP) [2] dataset. In the this dataset, there are 9 languages, i.e., ar, bn, en, fi, id, ko, ru, sw, te.

For NLI, we use the Cross-lingual Natural Language Inference corpus [3] and MultiNLI training data [4]. In this XNLI dataset, there are 15 languages, i.e., ar, bg, de, el, en, es, fr, hi, ru, sw, th, tr, ur, vi, zh.

Besides, our experiments are conducted on virtual machines of Microsoft Azure cloud, which has Nvidia V100 Tensor Core GPUs with 32 GB graphics memory.

References

[1] Xiaoman Pan, Boliang Zhang, Jonathan May, Joel Nothman, Kevin Knight, and Heng Ji. Cross-lingual name tagging and linking for 282 languages. In *ACL*, 2017.

[2] Jonathan H. Clark, Jennimaria Palomaki, Vitaly Nikolaev, Eunsol Choi, Dan Garrette, Michael Collins, and Tom Kwiatkowski. Tydi QA: A benchmark for information-seeking question answering in typologically diverse languages. *Trans. Assoc. Comput. Linguistics*, 8:454–470, 2020.

[3] Alexis Conneau, Ruty Rinott, Guillaume Lample, Adina Williams, Samuel R. Bowman, Holger Schwenk, and Veselin Stoyanov. XNLI: evaluating cross-lingual sentence representations. In *EMNLP*, 2018.

[4] Adina Williams, Nikita Nangia, and Samuel R. Bowman. A broad-coverage challenge corpus for sentence understanding through inference. In *NAACL-HLT*, 2018.