P-Tuning v2: Prompt Tuning Can Be Comparable to Fine-tuning Universally Across Scales and Tasks

Xiao Liu[1,2*], **Kaixuan Ji**[1*], **Yicheng Fu**[1*], **Weng Lam Tam**[1], **Zhengxiao Du**[1,2], **Zhilin Yang**[1,3†], **Jie Tang**[1,2†]
[1]Tsinghua University, KEG [2]Beijing Academy of Artificial Intelligence (BAAI)
[3]Shanghai Qi Zhi Institute
{liuxiao21,jkx19,fyc19}@mails.tsinghua.edu.cn

Abstract

Prompt tuning, which only tunes continuous prompts with a frozen language model, substantially reduces per-task storage and memory usage at training. However, in the context of NLU, prior work reveals that prompt tuning does not perform well for normal-sized pretrained models. We also find that existing methods of prompt tuning cannot handle hard sequence labeling tasks, indicating a lack of universality. We present a novel empirical finding that properly optimized prompt tuning can be universally effective across a wide range of model scales and NLU tasks. It matches the performance of finetuning while having only 0.1%-3% tuned parameters. Our method P-Tuning v2 is an implementation of Deep Prompt Tuning (Li and Liang, 2021; Qin and Eisner, 2021) optimized and adapted for NLU. Given the universality and simplicity of P-Tuning v2, we believe it can serve as an alternative to finetuning and a strong baseline for future research.[1]

1 Introduction

Pretrained language models (Radford et al., 2019; Devlin et al., 2018; Yang et al., 2019; Raffel et al., 2019) improve performance on a wide range of natural language understanding (NLU) tasks. A widely-used method, **fine-tuning**, updates the entire set of model parameters for a target task. While fine-tuning obtains good performance, it is memory-consuming during training because gradients and optimizer states for all parameters must be stored. Moreover, keeping a copy of model parameters for each task during inference is inconvenient since pre-trained models are usually large.

Prompting, on the other hand, freezes all parameters of a pre-trained model and uses a natural language prompt to query a language model (Brown et al., 2020). For example, for sentiment analysis, we can concatenate a sample (e.g., "Amazing movie!") with a prompt "This movie is [MASK]" and ask the pre-trained language model to predict the probabilities of masked token being "good" and "bad" to decide the sample's label. Prompting requires no training at all and stores one single copy of model parameters. However, discrete prompting (Shin et al., 2020; Gao et al., 2020) can lead to suboptimal performance in many cases compared to fine-tuning.

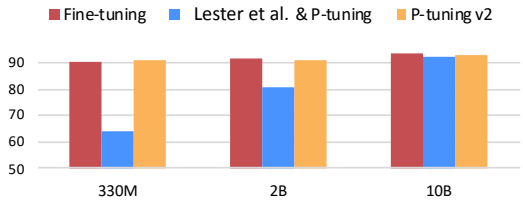

Figure 1: Average scores on RTE, BoolQ and CB of SuperGLUE dev. With 0.1% task-specific parameters, P-tuning v2 can match fine-tuning across wide scales of pre-trained models, while Lester et al. (2021) & P-tuning can make it conditionally at 10B scale.

Prompt tuning[2] is an idea of tuning only the continuous prompts. Specifically, Liu et al. (2021); Lester et al. (2021) proposed to add trainable continuous embeddings (also called continuous prompts) to the original sequence of input word embeddings. Only the continuous prompts are updated during training. While prompt tuning improves over prompting on many tasks (Liu et al., 2021; Lester et al., 2021; Zhong et al., 2021), it still underperforms fine-tuning when the model size is not large, specifically less than 10 billion parameters (Lester et al., 2021). Moreover, as shown in our experiments, prompt tuning performs poorly compared to fine-tuning on several hard sequence labeling tasks such as extractive question answer-

† corresponding to: Zhilin Yang (zhiliny@tsinghua.edu.cn) and Jie Tang (jietang@tsinghua.edu.cn)
* indicates equal contribution.
[1]Our code and data are released at https://github.com/THUDM/P-tuning-v2.
[2]We use "prompt tuning" to refer to a class of methods rather than a particular method.