Parameter Efficiency, Few Shot, Zero Shot, Prompting

Jonathan May

September 24, 2024

Preamble

These notes and previous notes on Pretrained LMs are possibly the most likely of all notes you'll get in class to go out of date. See that handout for details.

1 Parameter Efficiency

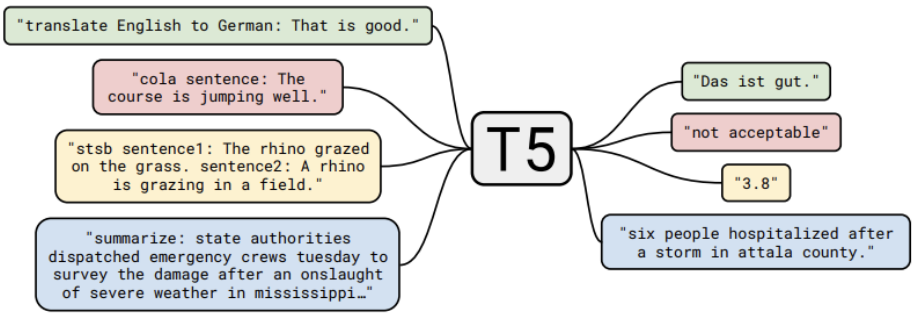
The models we've discussed so far follow the paradigm that, out of the box, they don't do too much, but when you expose them to some supervised data that is an exemplar of a task and fine-tune their parameters they can do the task when given more input data. One problem with this paradigm is that the base models are quite large, and then, when fine-tuned, you have another model that is as large as the base. If you have k tasks you have to store k copies of the fine-tuned base model. This is inefficient, so there have been efforts to allow the scaling to many tasks without exploding the number of models that have to be saved. This is an active area of research (as of this 2024 update), but here are a few interesting approaches to parameter efficiency.[1]

1.1 Multi-Task Learning

If you train a model to do more than one thing, then you implicitly are saving parameters.[2] T5 (referenced above) is one example of that approach. The pre-trained model is fine-tuned on many tasks all at once, each prepended with an instruction relevant to that task. T5 has since been further fine-tuned on more downstream tasks. A limitation of this approach is that the instructions are 'hard', i.e., you can do the tasks you're trained on, but it isn't clear you can do any other tasks.

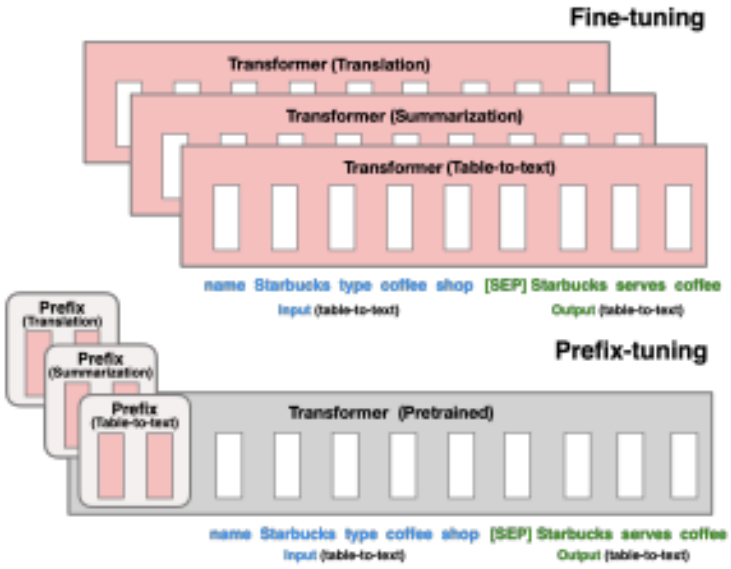
[1] Some of what follows from here: `https://github.com/allenai/acl2022-zerofewshot-tutorial`

[2] Note that the term *multi-task learning* is also applied to a case where you use a single model to do two different kinds of prediction at the same time. I'm abusing the term here.



1.2 Continuous Prefix Tuning

The idea behind prefix tuning[3] and a number of other related works is that, rather than imagining or declaring what the task prefix will be as in T5, it might be better to learn the prefix as well. Thus, we can imagine several otherwise unused tokens being learned when a pre-trained model is exposed to new task data; the rest of the model isn't modified! Then, instead of preceding your input with 'summarize' you precede it with '[TASK465]' or whatever that you've previously learned. You could do this for any number of tasks independently. There are a few versions of this; the one from [3] is shown below. The results on a few tasks are sometimes better than full fine-tuning but the goal is to reach parity despite only fine-tuning 0.1% of the parameters.

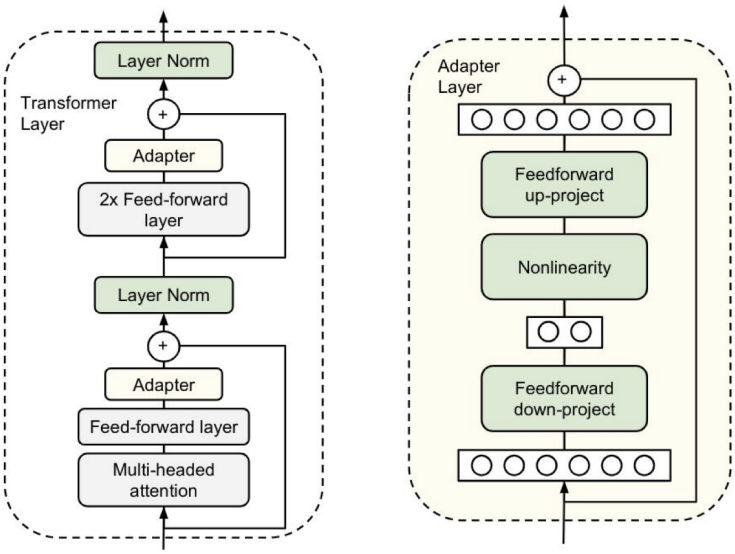


1.3 Adapters

Adapters [1] are kind of the same idea as prefix tuning but instead of adjoining to the left of the stack, they surround each layer of the transformer. The motivation for these came out slightly different from those of prefixes; before adapters, people were playing around with

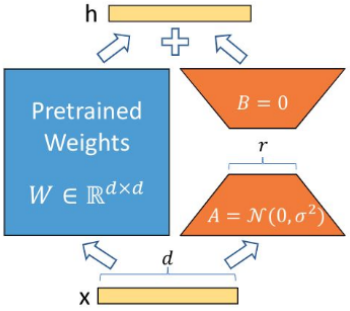
[3] https://aclanthology.org/2021.acl-long.353/

only fine-tuning some of the layers in a PTLM, and adapters were shown to be more efficient than this approach. A slightly larger footprint was claimed in the original adapters paper (3.6%) than by prefixes, but this is of course a hyperparameter. Below is a figure from the paper showing how adapters are added in. A goal of adapters had been to make a 'plug and play' approach to model specialization by using off-the-shelf adapters built for one task and combining them together; I haven't heard too much about this lately, probably because of zero-shot models changing the paradigm somewhat.



1.4 LoRA

Instead of adding parameters around the current stack why not add them *to* the stack? That is, given $d \times d$ weight matrix W (e.g. a Q, K, V matrix), let $W' = W + W_l$ and let $W_l = W_{l1} \times W_{l2}$ where W_{l1} is $d \times r$ and W_{l2} is $r \times d$, $r << d$. How much lower? Well, it varies, but values of 4 or 8 were used in the paper [2]. A 'shadow matrix' was learned for the W matrices (sometimes only a subset of them). As few as 0.02 parameters were learned (in the case of GPT-3) with very good results! Deep in the appendix of the paper the authors noted the combination of LoRA and prefix tuning worked very well!



Model&Method	# Trainable Parameters	WikiSQL Acc. (%)	MNLI-m Acc. (%)	SAMSum R1/R2/RL
GPT-3 (FT)	175,255.8M	**73.8**	89.5	52.0/28.0/44.5
GPT-3 (BitFit)	14.2M	71.3	91.0	51.3/27.4/43.5
GPT-3 (PreEmbed)	3.2M	63.1	88.6	48.3/24.2/40.5
GPT-3 (PreLayer)	20.2M	70.1	89.5	50.8/27.3/43.5
GPT-3 ($\text{Adapter}^{\text{H}}$)	7.1M	71.9	89.8	53.0/28.9/44.8
GPT-3 ($\text{Adapter}^{\text{H}}$)	40.1M	73.2	**91.5**	53.2/29.0/45.1
GPT-3 (LoRA)	4.7M	73.4	**91.7**	**53.8/29.8/45.9**
GPT-3 (LoRA)	37.7M	**74.0**	**91.6**	53.4/29.2/45.1

2 Zero-Shot Models

2.1 GPT-2

OpenAI, presumably not happy with being overshadowed by BERT, but not really interested in branding any better, released the paper "Language Models are Unsupervised Multitask Learners"[4] that described GPT-2 in February 2019. It was trained on 8m documents and 40GB of text sourced from outbound links from Reddit. The biggest model, at 1.5b parameters, exceeded SOTA performance considerably on a variety of LM sets even though it was not adapted to them. It was even able to do well (but not SOTA) on machine translation, question answering, and summarization tasks just by passing in natural language sequence prompts that attempted to elicit the kind of task response desired (e.g. inputting an article, then "TL;DR", then allowing the system to generate). The thing that really made the news was the (cherry-picked) 'unicorn' story. The other thing that made the news was OpenAI's refusal to let nearly anyone actually use the full system for the first year or so, claiming it was 'too dangerous.' So there was a lot of skepticism. But eventually, the models were released, and the claims were, in fact, justified.

Language Models are Unsupervised Multitask Learners

	LAMBADA (PPL)	LAMBADA (ACC)	CBT-CN (ACC)	CBT-NE (ACC)	WikiText2 (PPL)	PTB (PPL)	enwik8 (BPB)	text8 (BPC)	WikiText103 (PPL)	1BW (PPL)
SOTA	99.8	59.23	85.7	82.3	39.14	46.54	0.99	1.08	18.3	**21.8**
117M	**35.13**	45.99	**87.65**	**83.4**	**29.41**	65.85	1.16	1.17	37.50	75.20
345M	**15.60**	55.48	**92.35**	**87.1**	**22.76**	47.33	1.01	**1.06**	26.37	55.72
762M	**10.87**	**60.12**	**93.45**	**88.0**	**19.93**	**40.31**	**0.97**	**1.02**	22.05	44.575
1542M	**8.63**	**63.24**	**93.30**	**89.05**	**18.34**	**35.76**	**0.93**	**0.98**	**17.48**	42.16

Table 3. Zero-shot results on many datasets. No training or fine-tuning was performed for any of these results. PTB and WikiText-2 results are from (Gong et al., 2018). CBT results are from (Bajgar et al., 2016). LAMBADA accuracy result is from (Hoang et al., 2018) and LAMBADA perplexity result is from (Grave et al., 2016). Other results are from (Dai et al., 2019).

[4] `https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf`

Context (human-written): In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.
GPT-2: The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science. Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved. Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow. Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez. Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns. While examining these bizarre creatures the scientists discovered that the creatures also spoke some fairly regular English. Pérez stated, "We can see, for example, that they have a common 'language,' something like a dialect or dialectic." Dr. Pérez believes that the unicorns may have originated in Argentina, where the animals were believed to be descendants of a lost race of people who lived there before the arrival of humans in those parts of South America. While their origins are still unclear, some believe that perhaps the creatures were created when a human and a unicorn met each other in a time before human civilization. According to Pérez, "In South America, such incidents seem to be quite common." However, Pérez also pointed out that it is likely that the only way of knowing for sure if unicorns are indeed the descendants of a lost alien race is through DNA. "But they seem to be able to communicate in English quite well, which I believe is a sign of evolution, or at least a change in social organization," said the scientist.

Table 13. Conditional generation on an out-of-distribution context by GPT-2. Cherry pick of 10 samples generated with $k = 40$.

It's worth pausing to describe the two (ish) approaches that are being proposed here and why they are now so cool. The main reason they are cool is that the model parameters aren't ever changed, so manipulation of behavior occurs only at inference time.

1. **In-Context Learning** or demonstration based learning. Several (two? three? ten?) examples of desired input-output behavior are given, then one or several inputs without an output are given and the model produces outputs that follow the pattern.

2. **Instruction Learning**. A natural language description of what is wanted is produced, then an input is given, and the model produces outputs that are responsive to the instruction.

In practice, both are used together. There are also questions about the specific value of each component (see below).

2.2 GPT-3

In 2020 OpenAI released "Language Models are Few-Shot Learners"[5] which heralded the release of GPT-3 in a paper that was 72 pages long (GPT-2 paper was 24 pages long) and that described a model with 175b parameters. It was trained on 570GB of text (500b tokens) from the Common Crawl, their previous data set, and some other 'high quality' data sets. When prompted with only a few examples (and sometimes with none at all) it outperformed SOTA on a wide variety of tasks, including common sense reasoning tasks, machine translation, question answering, and others. This time OpenAI set up an API and web interface so that researchers and others could use the models. The results have been really excellent though sometimes care is needed to prompt appropriately.

Model Name	n_{params}	n_{layers}	d_{model}	n_{heads}	d_{head}	Batch Size	Learning Rate
GPT-3 Small	125M	12	768	12	64	0.5M	6.0×10^{-4}
GPT-3 Medium	350M	24	1024	16	64	0.5M	3.0×10^{-4}
GPT-3 Large	760M	24	1536	16	96	0.5M	2.5×10^{-4}
GPT-3 XL	1.3B	24	2048	24	128	1M	2.0×10^{-4}
GPT-3 2.7B	2.7B	32	2560	32	80	1M	1.6×10^{-4}
GPT-3 6.7B	6.7B	32	4096	32	128	2M	1.2×10^{-4}
GPT-3 13B	13.0B	40	5140	40	128	2M	1.0×10^{-4}
GPT-3 175B or "GPT-3"	175.0B	96	12288	96	128	3.2M	0.6×10^{-4}

Table 2.1: Sizes, architectures, and learning hyper-parameters (batch size in tokens and learning rate) of the models which we trained. All models were trained for a total of 300 billion tokens.

2.3 Chat GPT and beyond

With somewhat less fanfare, in March 2022 OpenAI released 'Training language models to follow instructions with human feedback'[6] and replaced its GPT-3 models with these. Even the smallest (1.3b param) models were shown to be preferred to GPT-3 175b by users. They were fine-tuned to respond better to user prompts using 'Reinforcement Learning with Human Feedback' (which we will cover in a separate lecture). The authors also claim these models are less toxic than previous models.

In late November 2022, OpenAI released Chat GPT, a chatbot interface that wrapped this enhanced GPT-3 in a free-to-use and widely available interface. The promotion of and response to ChatGPT was very very loud. Within a few months seemingly everyone, even beyond the tech world, was aware of the technology, surprised at what it could do, and possibly scared of it. This kicked off an arms race among leading tech companies to build their own models (which started being called 'large language models' despite the term being fairly vapid).

2.4 Why/how does in-context learning work?

(From `https://github.com/allenai/acl2022-zerofewshot-tutorial`)

Some views (this is an area of active study, though):

[5] `https://arxiv.org/abs/2005.14165`

[6] `https://arxiv.org/abs/2203.02155`

- Demonstrations do not teach a new task; instead, they allow the 'locating' of an already-learned task during pretraining (Reynolds & McDonell, 2021)
- LMs do not exactly understand the meaning of their prompt (Webson & Pavlick, 2021)
- Demonstrations are about providing a latent concept so that LM generates coherent next tokens (Xie et al. 2022)
- In-context learning performance is highly correlated with term frequencies during pretraining (Razeghi et al. 2022)
- LMs do not need input-label mapping in demonstrations, instead, they use the specification of the input & label distribution separately (Min et al. 2022)
- Data properties lead to the emergence of few-shot learning (burstiness, long-tailedness, many-to-one or one-to-many mappings, a Zipfian distribution) (Chan et al. 2022)

An interesting finding from all of this (from Min et al. 22[7]) is that you can prompt with examples but *random* answers and get basically the same results.

[7] `https://arxiv.org/abs/2202.12837` via the aforementioned tutorial

References

[1] Neil Houlsby et al. "Parameter-Efficient Transfer Learning for NLP". In: *CoRR* abs/1902.00751 (2019). arXiv: 1902.00751. URL: http://arxiv.org/abs/1902.00751.

[2] Edward J. Hu et al. "LoRA: Low-Rank Adaptation of Large Language Models". In: *CoRR* abs/2106.09685 (2021). arXiv: 2106.09685. URL: https://arxiv.org/abs/2106.09685.

[3] Xiang Lisa Li and Percy Liang. "Prefix-Tuning: Optimizing Continuous Prompts for Generation". In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 4582–4597. DOI: 10.18653/v1/2021.acl-long.353. URL: https://aclanthology.org/2021.acl-long.353.