Chapter 17. Advanced Transformer Techniques

In Chapters 15 and 16, we built all kinds of transformers, from classifiers, translators and chatbots, to vision and multimodal transformers. While transformers are incredibly versatile and powerful, they are far from perfect.

Firstly, transformers are computationally expensive. Luckily, many techniques have been developed to speed up transformers of any size:

- To accelerate decoding, we will use key/value caching, multi-token prediction, and speculative decoding.
- To boost multi-head attention (MHA), which is one of the most computationally expensive components of transformers, we will look at sparse attention, approximate attention, weight sharing, and FlashAttention.
- To speed up gigantic transformers of up to trillions of parameters, we will discuss Mixture of Experts (MoE).
- To train large transformers efficiently, we will discuss data parallelism, model parallelism, gradient checkpointing, sequence packing, zero redundancy optimizer (ZeRO), and *parameter-efficient fine-tuning* (PEFT) using adapters such as *Low-Rank Adaptation* (LoRA).

TIP

Another way to speed up a transformer is to make it smaller. This is discussed in Appendix B.

Secondly, transformers struggle with long sequences. This is in large part because of the quadratic attention problem, which can be alleviated using the MHA-acceleration techniques listed above. However, transformers also struggle with very long sequences for another reason: their accuracy drops because they haven't seen enough long sequences during training. Luckily, modern positional encoding techniques such as Relative Position Encodings (RPE), Rotary Position Embeddings (RoPE), and Attention with Linear Biases (ALiBi) can help transformers extrapolate to longer sequence lengths.

Lastly, we will discuss state-space models (SSMs) such as Mamba or Hyena, which can process extremely long sequences, scaling linearly rather than quadratically. SSMs are not actually transformers, as they don't even use MHA; instead, you can think of them as optimized linear RNNs. They offer an efficient alternative to transformers for very long inputs (e.g., genomics).

That's quite a lot of techniques to cover, and they are fairly advanced, so you can safely skip this chapter for now if you are new to transformers, and come back later whenever needed. This is why this chapter is online-only, available at *https://homl.info/att*, to leave room for the other chapters.