Parameter-efficient fine-tuning (PEFT)

- High-level: we want to avoid modifying most of the pretrained model’s parameters during fine-tuning.

- Prompting: requires adjusting zero params to solve a downstream task.

What is the sentiment of the below sentence? Answer w/ either “pos” or “neg”.
Input sentence: 
Output: pos

- Prompt engineering

- Limitations:
  - Hard to solve very complex reasoning/understanding tasks
  - Requirements for the pretrained model are immense
    - Huge-scale pretraining
    - High quality large scale instruction tuning
  - RLHF requires access to very expensive human proof datasets
Review of full model finetuning:

- Input: 
  - Pretrained decoder
  - This movie is good (EOS)

Predict positive

Prompt tuning (Lester et al., 2022)

- Input:
  - Pretrained decoder
  - This movie is good

Predict positive

Update: keep all pretrained params frozen,
only do

\[
e_1_{\text{new}} = e_1_{\text{old}} - \eta \frac{dl}{de_1},
\]

\[
e_2_{\text{new}} = \ldots
\]
LoRA (low-rank adaptation):

\[ h \approx f(Wx) \frac{dL}{dW} \]

\( W \) is an \( m \times n \) matrix
\( \frac{dL}{dW} \) is also \( m \times n \)

\[ W_{\text{new}} = W_{\text{old}} - h \frac{dL}{dW} \]

having two low-rank matrices \( A \) and \( B \)

\( r = \text{rank parameter} \)
\( \text{want } r \ll cccccc m, n \)

product \( AB^T \), \( m \times n \)

in LoRA:

\[ h = f((W_{\text{old}} + AB^T)x) \]
we compute \( \frac{dl}{dA} \) and \( \frac{dl}{dB} \), much smaller than \( \frac{dl}{dw} \).

At the end of LoRA fine-tuning, we have a separate \( A, B \) for each tuned weight matrix.

\[
W_{\text{new}} = W_{\text{LoRA}} AB^T \quad f(W_{\text{new}} x)
\]

\( \text{QLoRA: quantized LoRA} \)

normal models: FP32

\( \downarrow \)

4 bit, 8 bit integer quantization