LoRa

A Preprint

During full fine-tuning, a model with pre-trained weights Φ_0 is updated to $\Phi_0 + \Delta\Phi$ by maximizing the conditional language modeling objective:

$$\max_{\Phi} \sum_{(x,y)\in\mathcal{Z}} \sum_{t=1}^{|y|} \log P_{\Phi}(y_t \mid x, y_{<t}) \tag{1}$$

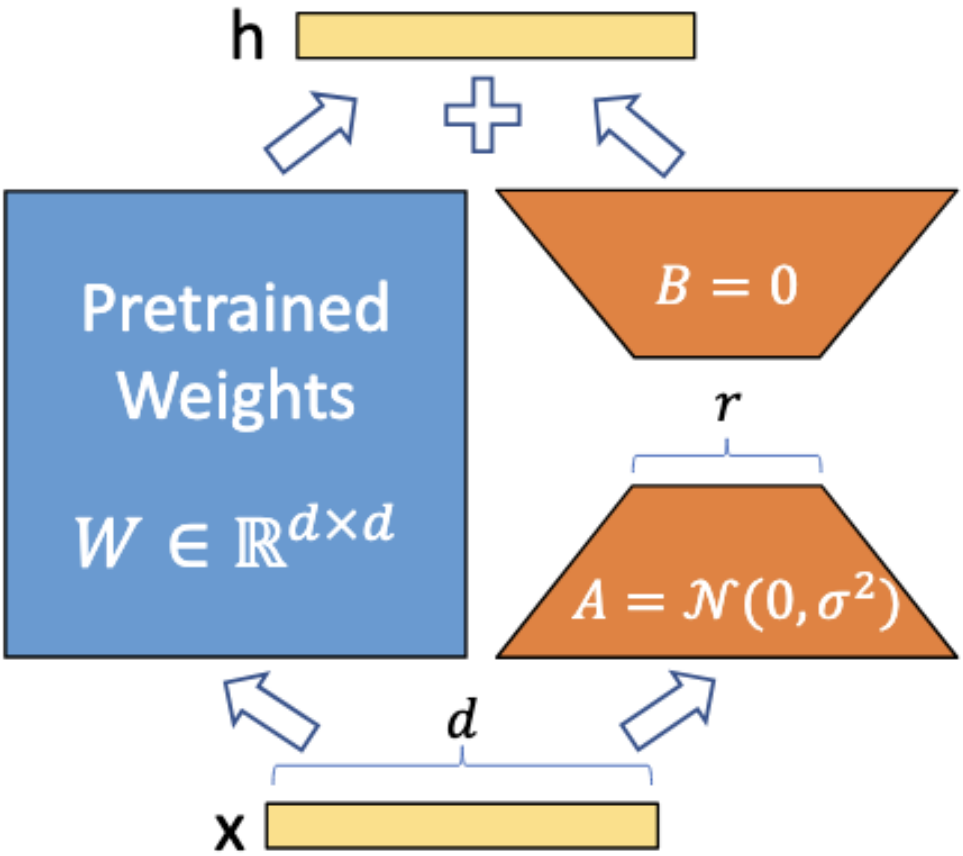
A major drawback of full fine-tuning is that each downstream task requires learning a separate parameter set $\Delta\Phi$ of the same size as Φ_0, i.e., $|\Delta\Phi| = |\Phi_0|$. For large models (e.g., GPT-3 with $|\Phi_0| \approx 175$B), this becomes impractical in terms of storage and deployment.

To address this, a more parameter-efficient approach is considered: the task-specific increment $\Delta\Phi$ is expressed as a function of a much smaller parameter set Θ, where $|\Theta| \ll |\Phi_0|$, i.e., $\Delta\Phi = \Delta\Phi(\Theta)$. The optimization then becomes:

$$\max_{\Theta} \sum_{(x,y)\in\mathcal{Z}} \sum_{t=1}^{|y|} \log P_{\Phi_0+\Delta\Phi(\Theta)}(y_t \mid x, y_{<t}) \tag{2}$$

Subsequent sections introduce a low-rank representation of $\Delta\Phi$, aimed at reducing both compute and memory costs. For models like GPT-3, the number of trainable parameters $|\Theta|$ can be as low as 0.01% of $|\Phi_0|$.

The main idea that authors present is replacing $\Delta W x$ with BAx. For A gaussian init is used, B is init as 0 so $\Delta W = BA$ at the beginning of the training. Original model is kept frozen while difference from original model is used. Using low rank decomposition amount of parameters is reduces. However, the problem is that if multiple different inputs for different tasks are in one batch - you can't easily create a single merged $W^{'}$ which works for entire batch. It is possible to dynamically choose different modules however it comes out in additional inference latency.


h
Pretrained Weights
$W \in \mathbb{R}^{d \times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x