

DoRA vs LoRA:

DoRA: DoRA: Weight-Decomposed Low-Rank Adaptation, a potential successor to LoRA (low-rank adaptation).

LoRA is probably the most widely used parameter-efficient finetuning method for LLMs and vision transformers, and DoRA can be seen as an improvement or extension of LoRA that is built on top of it.

A brief LoRA recap: Assuming we have pretrained model weights W , LoRA uses low-rank matrices to approximate weight changes ΔW . I.e., in regular finetuning we have $W' = W + \Delta W$, and in LoRA, we approximate ΔW with BA.

Now, the DoRA method first decomposes the pretrained weight matrix into a magnitude vector (m) and a directional matrix (V). Then, it takes the directional matrix V and applies standard LoRA to it, i.e., $W' = m (V + \Delta V)/\text{norm} = m (W + BA)/\text{norm}$.

The motivation for developing this method is based on analyzing and comparing the LoRA and full finetuning learning patterns. They found that LoRA either increases or decreases magnitude and direction updates proportionally but seems to lack the capability of making only subtle directional changes as found in full finetuning. Hence, the researchers propose the decoupling of magnitude and directional components. In other words, their DoRA method aims to apply LoRA only to the directional component (while also allowing the magnitude component to be trained separately.)

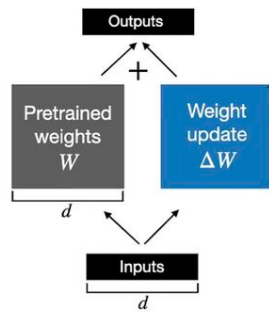
Note that introducing the magnitude vector m in DoRA adds 0.01% more parameters than standard LoRA. However, across both LLM and vision transformer benchmarks, they found that DoRA even outperforms LoRA if the DoRA rank is halved, i.e., when DoRA only uses half the parameters of regular LoRA.

Overall, I am actually quite impressed by the results and need to toy with this method in practice. It should not be too big of a lift to upgrade a LoRA implementation to DoRA.

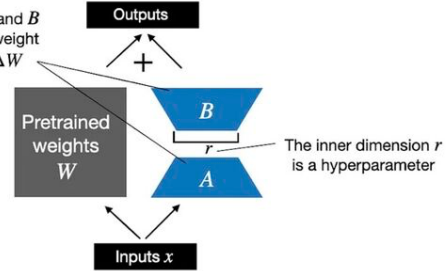
Paper: <https://lnkd.in/gihA-tyV>

Regular LoRA

Weight update in **regular finetuning**

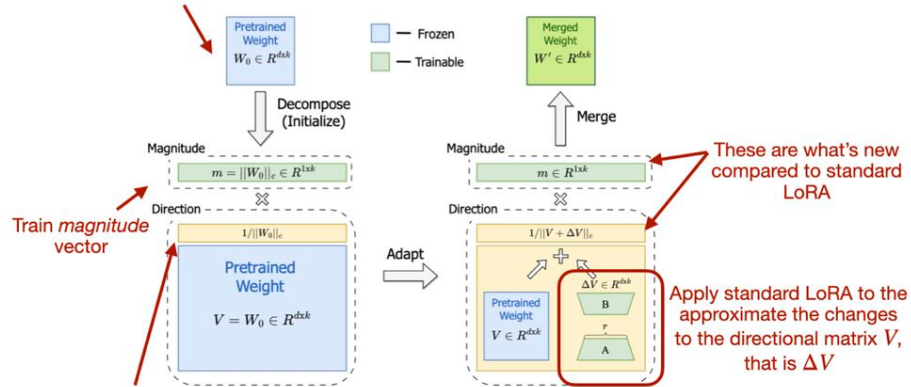


Weight update in **LoRA**



Step 1, decompose the pretrained weight matrix into magnitude vector m and directional matrix V

DoRA

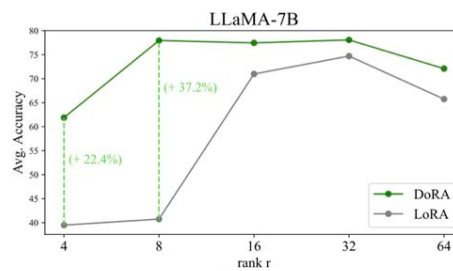


where the decomposition is

$$W_0 = m \frac{V}{\|V\|_c} = \|W_0\|_c \frac{W_0}{\|W_0\|_c}$$

hence the $1/\|W_0\|_c$

Step 2, finetune model with LoRA



Not only does DoRA perform better overall, but it is also more robust to changes in the rank parameter