LoRA: Low Rank Adaptation of LLM

Fine tuning means training a pre trained network on new data to improve its performance

on specific task. But the problem with full fine tuning is that it is computationally expensive.

Storage requirements for checkpoints are expensive.

Hence we introduce LoRA (Low Rank Adaptation of LLM)

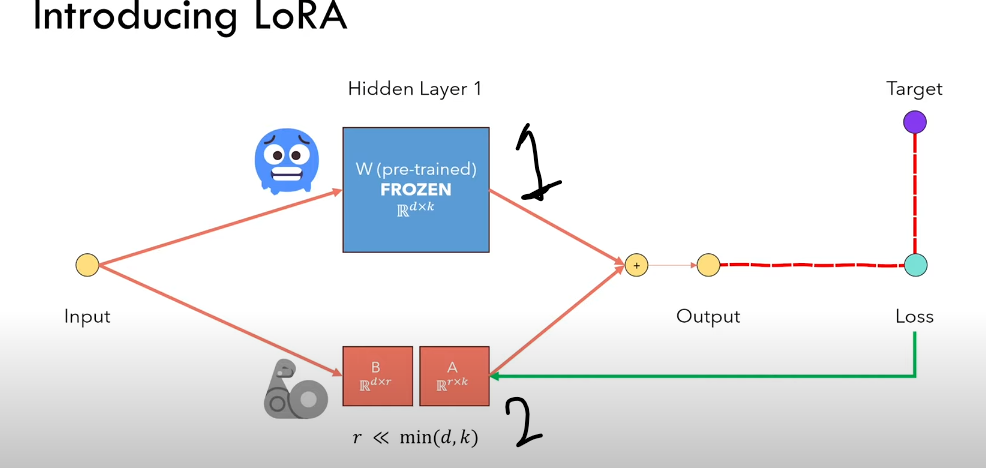
We freeze the model weights. Let's call it matrix 1.

We divide the model weight matrix into 2 parts, where original matrix d\*k is broken into d\*r and r\*k

where r<< min(d,k). Let's call it matrix 2.

We then add matrix 1 and matrix 2

We backpropagate the loss only through matrix 2.



Benefits?

* Less parameters to train
* Faster backpropagation
* We can easily switch between 2 models
* We don’t need to care about the weight matrix of the entire model since it is frozen as well as the optimizer states

Pretrained models have very low intrinsic dimension. In other words there exists a low dimension reparametrized that is as effective for fine tuning as full parameter space.

Rank of a matrix is not necessarily equal to the dimension of the matrix but equal to the number of linearly independent column. Hence, we use the method of matrix decomposition.

In the decomposition matrix r is the hyperparameter.