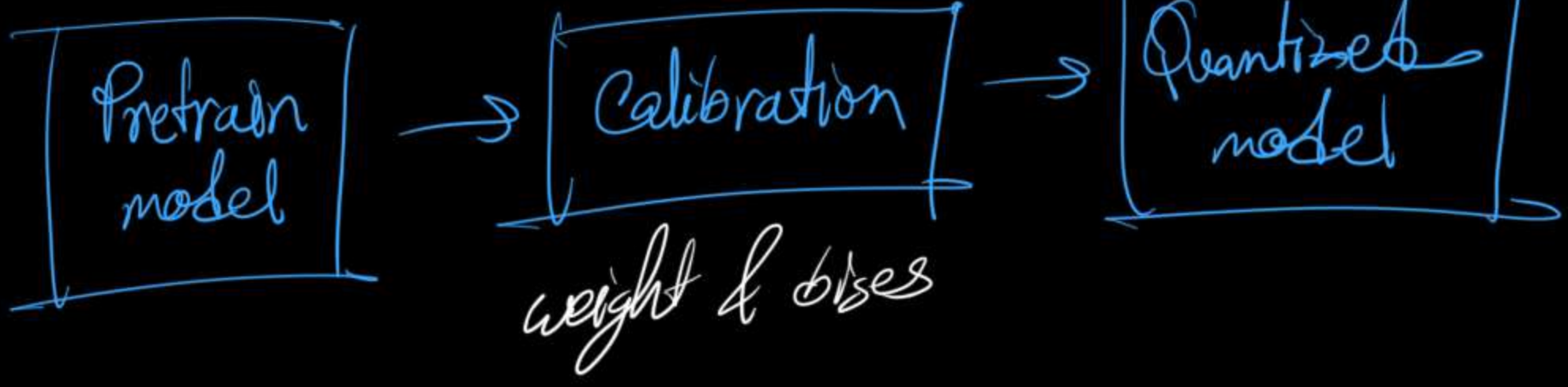
Quantization

Means converting the weights & biases from float 32 to float 16 or int 8.

That means making the model smaller & in this way we can reduce the computation.

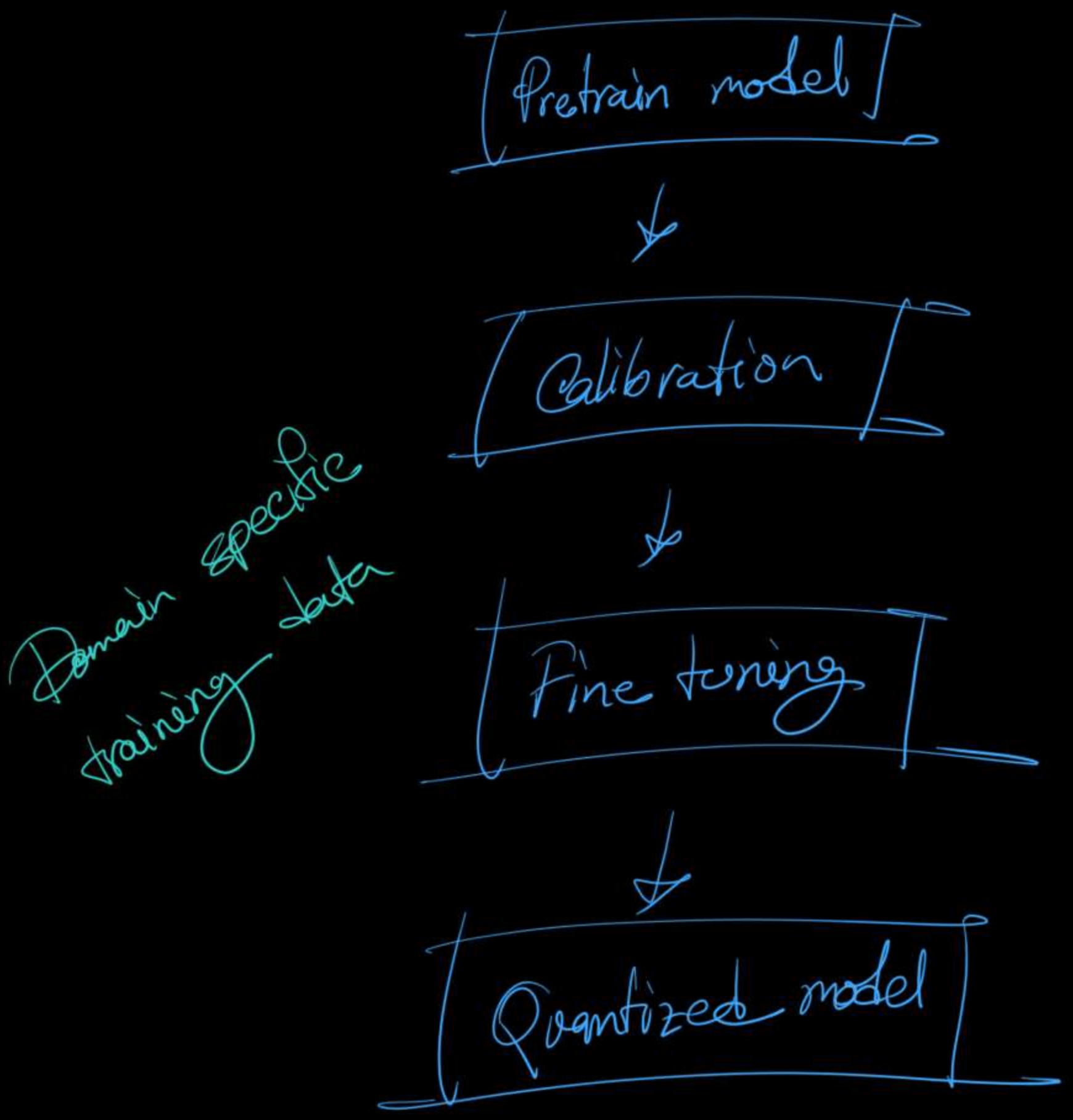
2 types of Quantization

1) Post training quantization



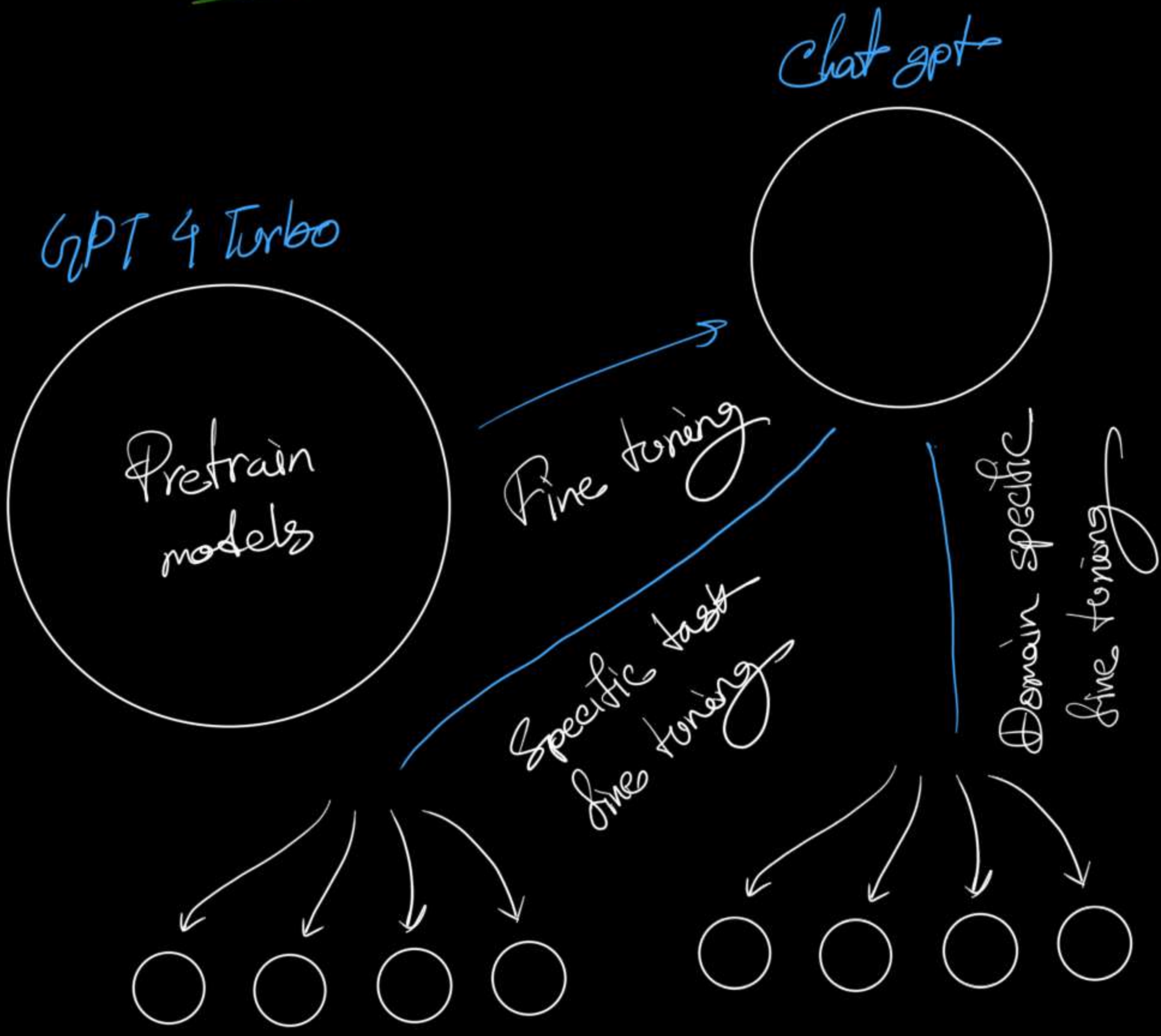
In this case the information gets lost in the way & the accuracy decreases.

Quantization aware training



We use this technique so that we don't loose information along the way.

LoRA / QLoRA

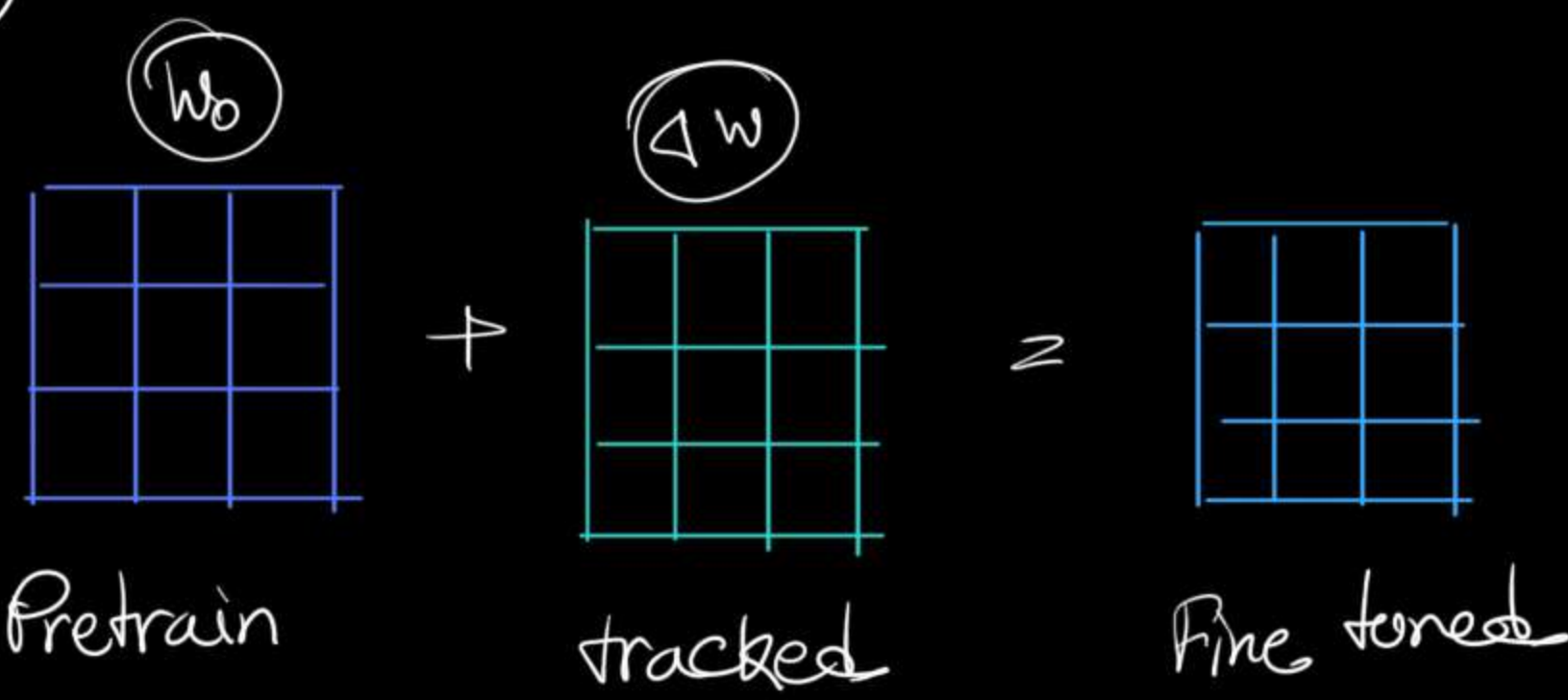


- Full Parameter FT
- Update all weights & biases
- Resource constraint

To solve the resource problem came LoRA & QLoRA

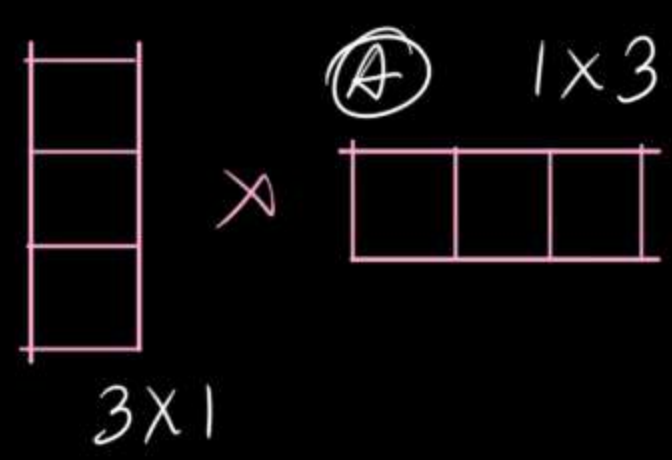
What LoRA Do?

- Instead of updating weights, it tracks changes of the new weights based on fine tuning.



Matrix Decomposition

based on Rank



we can get 9 parameters from just 6

$$W_0 + \Delta W = W_0 + BA$$

Number of Trainable Parameters

Method	Hyperparameters	# Trainable Parameters
Fine-Tune	-	175B
PrefixEmbed	$l_p = 32, l_i = 8$	0.4 M
	$l_p = 64, l_i = 8$	0.9 M
	$l_p = 128, l_i = 8$	1.7 M
	$l_p = 256, l_i = 8$	3.2 M
	$l_p = 512, l_i = 8$	6.4 M
PrefixLayer	$l_p = 2, l_i = 2$	5.1 M
	$l_p = 8, l_i = 0$	10.1 M
	$l_p = 8, l_i = 8$	20.2 M
	$l_p = 32, l_i = 4$	44.1 M
	$l_p = 64, l_i = 0$	76.1 M
Adapter^{H}	$r = 1$	7.1 M
	$r = 4$	21.2 M
	$r = 8$	40.1 M
	$r = 16$	77.9 M
	$r = 64$	304.4 M
LoRA	$r_v = 2$	4.7 M
	$r_q = r_v = 1$	4.7 M
	$r_q = r_v = 2$	9.4 M
	$r_q = r_k = r_v = r_o = 1$	9.4 M
	$r_q = r_v = 4$	18.8 M
	$r_q = r_k = r_v = r_o = 2$	18.8 M
	$r_q = r_v = 8$	37.7 M
	$r_q = r_k = r_v = r_o = 4$	37.7 M
	$r_q = r_v = 64$	301.9 M
	$r_q = r_k = r_v = r_o = 64$	603.8 M

Rank

If the model want to learn complex things the we use high Rank

Quantized LoRA

If the weights are stored into decomposed matrix in 16 bit, it is converted into 4bit.