**Finetuning LLMs**:

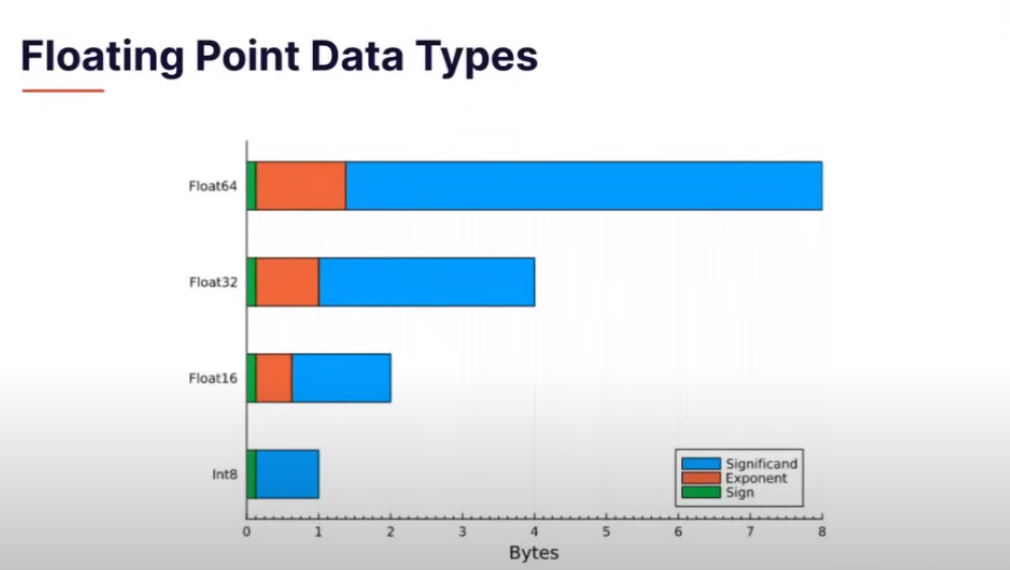
* The key points of focus:
  + How to do inference/finetuning on not very powerful servers as was raised in the previous week’s internal meeting
  + Some possible finetuning/ Meta Learning approaches for llms:
* References:
  + Llora paper : <https://arxiv.org/pdf/2106.09685.pdf>
  + Qlora paper : <https://arxiv.org/pdf/2305.14314.pdf>
  + Reference Video: <https://www.youtube.com/watch?v=g68qlo9Izf0>

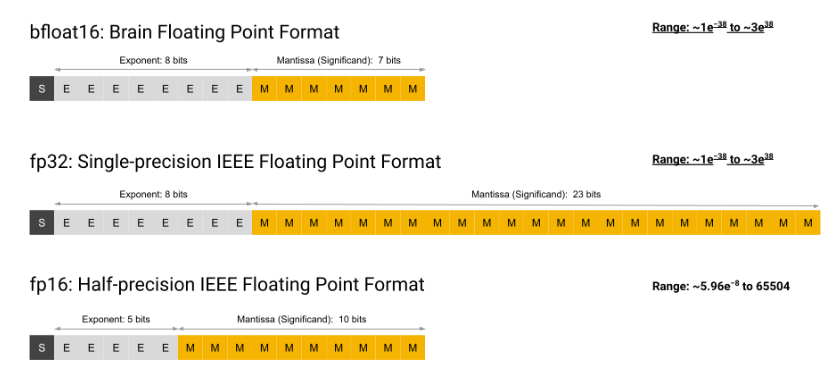
**Some Points to Note:**

* Codellama was trained using bfloat16 datatype.
* Pytorch by default loads weights in float32.
* For a codellama/ llama 7B model , assuming float32 is used everywhere
  + VRAM needed for model weights : 7\*4 = 28 GB.
  + Gradients would need 28 GB
  + optimizer like adam(twice the values as model) would need = 56GB.
  + Thus the total requirement is around 112GB VRAM(excluding activations) (DGX GPU specs: NVIDIA A100-SXM4-80GB).
* Thus it seems it is hard to finetune this smallest model on normal servers. A few techniques that help to reduce the memory footprint are as below:

| Technique | Improvements |
| --- | --- |
| Half Precision | Model Parameters, Gradients, Activation |
| Quantization | Model Parameters, Gradients, Activation |
| Lora | Gradients, Optimizer |
| Gradient Accumulation | Gradients, Activation |
| Paged Optimizers | Optimizer |

**Half Precision:**

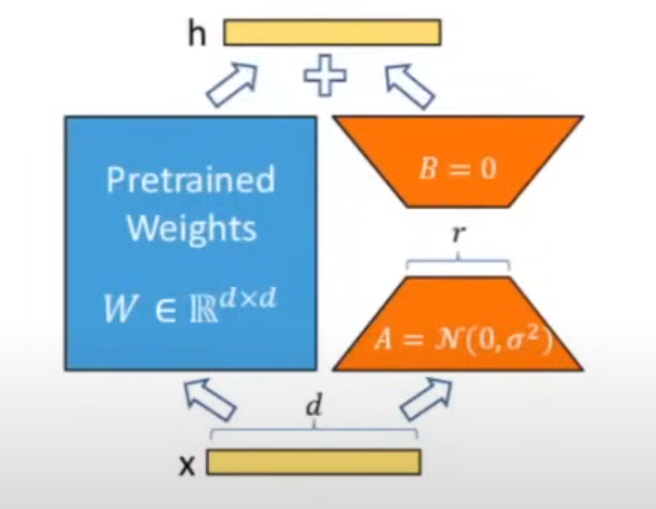


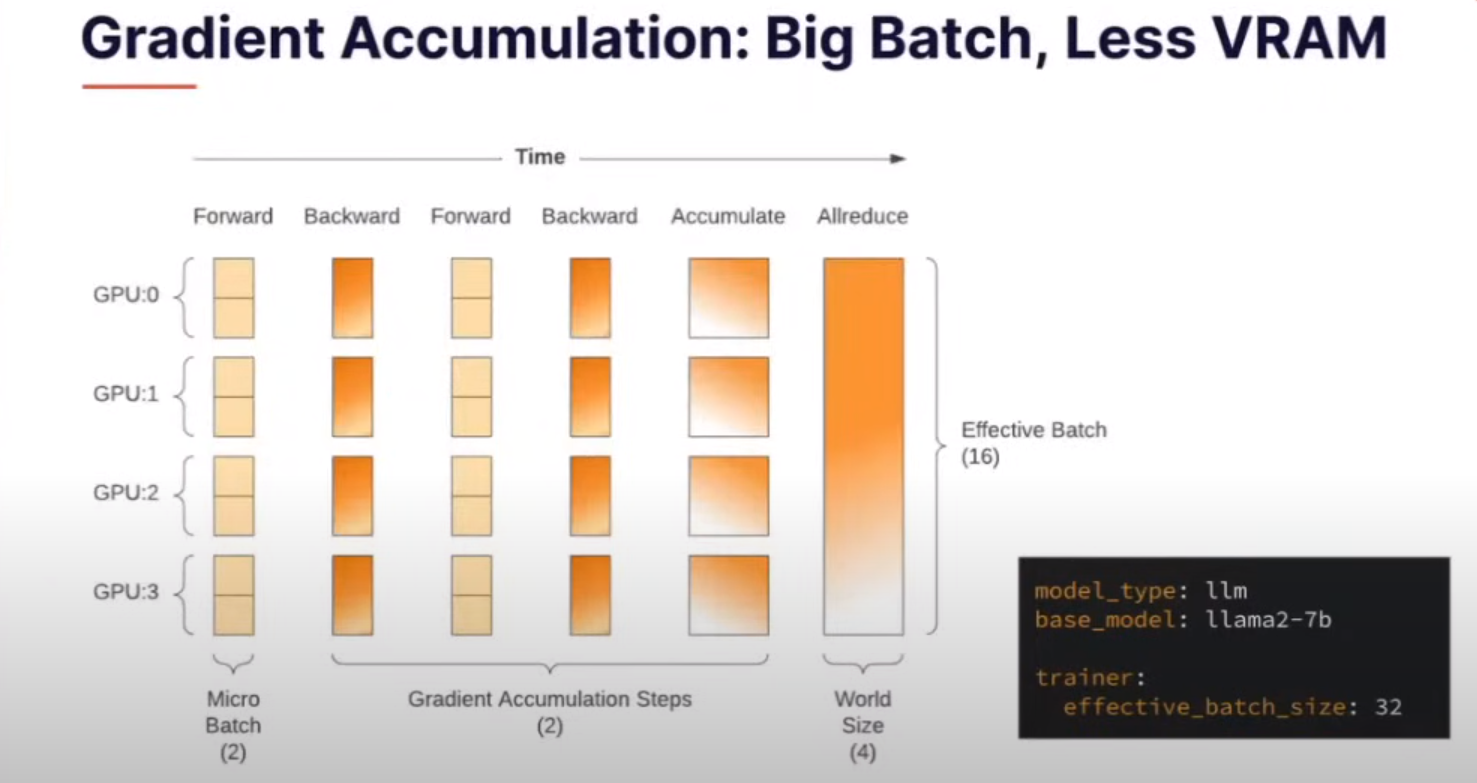


**Quantization:**

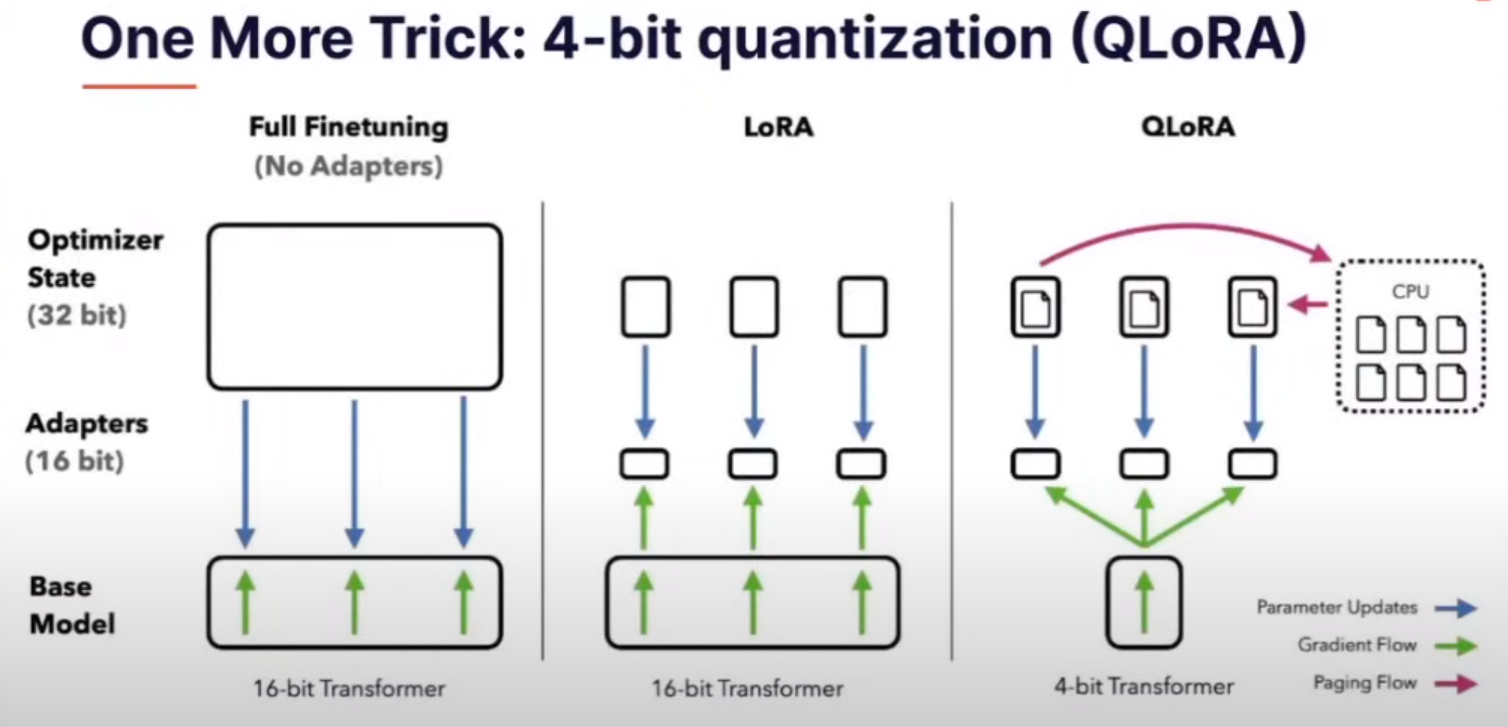


**Lora:**



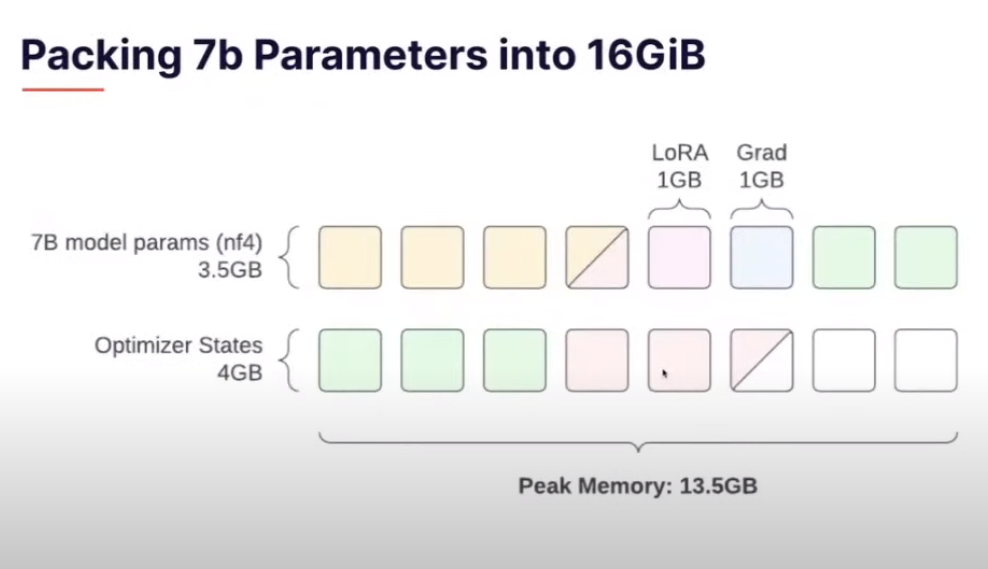
**Gradient Accumulation:** 

**QLora:**



**Footprint After these Optimizations**

* If we were to apply Qlora. Assuming gradients in bloat16 and optimizer in float32
  + The model weights would reduce to 7\*0.5 = 3.5 GB.
  + For code llama n is typically 4096 for matrices. Thus typical values of r=16 and using float16 for the lora weights the required memory would be around (7)\*(32/4096)\*2 which is nearly 0.1 GB . But lets assume for conservativeness (1GB) is needed for the Lora weighs.
  + Similarly 1GB for gradients of this Lora matrices.
  + For optimizer it will take 1\*2\*2=4GB.
  + Similarly Activations will take around 3 GB.
  + Total requirements is thus 12.5 GB from more then 112 GB initially.
* Since the size of the gradients is low now we can probably maintain multiple such lora heads in case of Meta Learning using Lora/ Qlora.



**- Code Llama Prompt Templates**

Prompt templates used for normal chat based/instruction finetuned inference in local huggingface code llama:

**Template :**

“<s>[INST] <<SYS>>\n{system\_prompt}\n<</SYS>>\n\n{user\_prompt1} [/INST] {model\_answer1} </s><s>[INST] {user\_prompt2} [/INST]”

**Example Chat Conversation:**

user\_prompt = “knock knock”

model\_prompt = “<s>[INST] <<SYS>>\nAlways answer as helpfully as possible.\n<</SYS>>\n\knock knock [/INST]”

model\_output = “[INST] <<SYS>>\nAlways answer as helpfully as possible.\n<</SYS>>\n\nknock knock [/INST] Who's there?”

model\_answer = “ Who's there?”

user\_prompt = “I am Margav”

model\_prompt = “<s>[INST] <<SYS>>\nAlways answer as helpfully as possible.\n<</SYS>>\n\nknock knock [/INST] Who's there? </s><s>[INST] I am Margav [/INST]”

model\_output = “[INST] <<SYS>>\nAlways answer as helpfully as possible.\n<</SYS>>\n\nknock knock [/INST] Who's there?<s>[INST] I am Margav [/INST] Hello Margav! It's nice to meet you. Is there anything you would like to talk about or ask? I'm here to help with any questions you may have.”

model\_answer = “ Hello Margav! It's nice to meet you. Is there anything you would like to talk about or ask? I'm here to help with any questions you may have.”