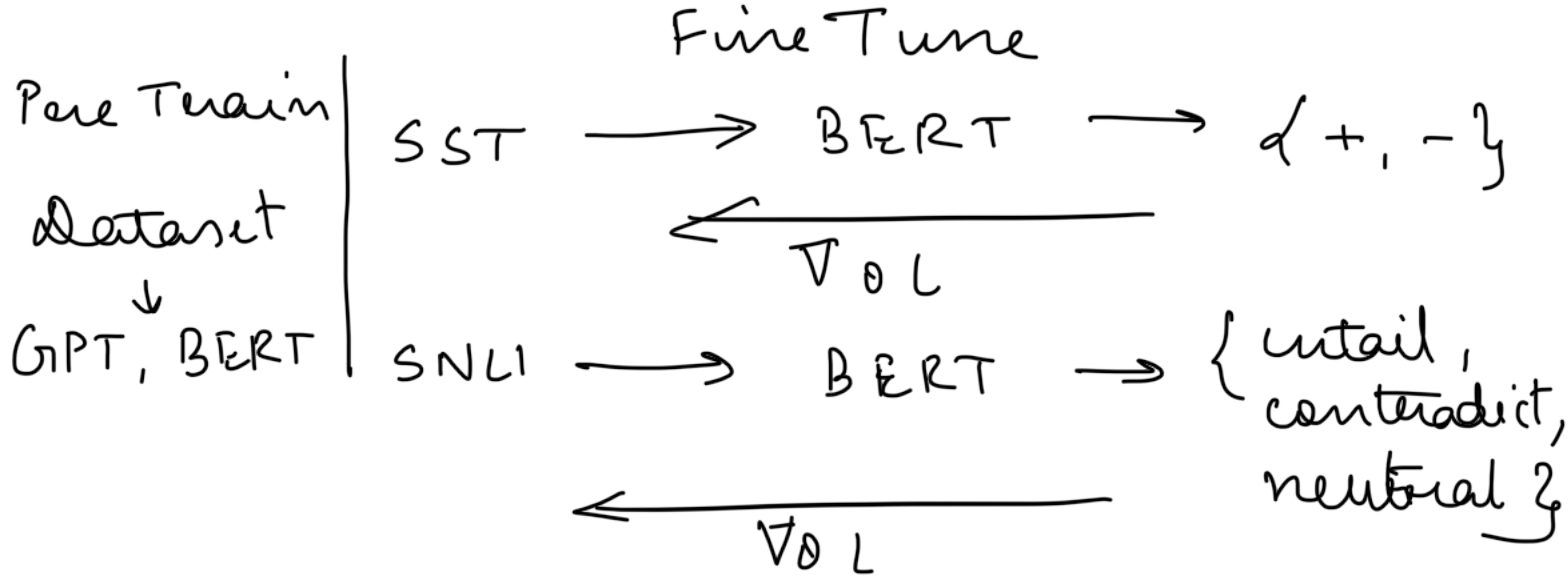
Week-4 DLP

L1 Task Specific Fine Tuning & It's limitations

- Trained the GPT-2 model with a CLM training objective.

$$\underline{\text{Min}} \quad L = -\sum_{D}\sum_{i=1}^{I} y_i (\log(\hat{y}_i))$$

- But, we have to refine it for the downstream tasks like classification, text generation & conversation.
- earlier, we had to fine tune for each downstream tasks.



- fine tuning → makes the model to adapt for downstream tasks. Initializing the params with the params learned by solving pre-trained objective
- At the input, add additional tasks based on type of downstream Task. At output replace the pre-training LM head with classification head of a linear layer.

- Our obj. to predict the label

$$\hat{y} = P(y | x_1 \dots x_m) = \text{softmax}(W_y h_L^m)$$

$$\text{Min. } L = -\sum_{(x,y)} \log(\hat{y}_i)$$

- Now, the pre-trained model acts as a feature extractor & the classificaⁿ head acts as a simple classifier. Or you have to train all the params of model known as full fine-tuning.
- GPT-2 small ⟶ 3 to 4 GB of GPU of batch size 1 to train the model.
- GPT-2 X large ⟶ > 22 or 24 GB GPU to train the model.
- So think of memory req. to fine-tune huge models?

L2 Emerging Abilities

- There are many approaches used so far to get memory req. to fine-tune LM.

  Additive, Selective, Adapters, Soft Prompts, Reparameterizaⁿ-based
- PEFT → Parameter efficient fine tuning

- In PEFT we have LORA, Glora, AdaLora are most used (+ quantiza^n)
- even for downstream fine tune you need lakhs of data points. Sometimes you don't have enough labelled samples. But what about humans, we don't see samples, right?
- ∴, we don't need supervised fine-tuning at all for most of the tasks.

Currently, $p(\text{output} \mid \text{input})$

Now, $p(\text{output} \mid \text{input}, \text{task})$

task is just an instruc^n in plain words to the input seq.

- In this setting, to get a good performance, we have to scale up both model size & data size.
- push the limits: 1.5B to 175B. The ability to learn to learn from few examples ↑ as model size increases.
- Since the model learns a new task from samples within the context window, this approach ⟶ in context learning.
- the adap^n occurs during inference, there is no need to share model wts. for

fine tuning. This method led to pavement of prompt engineering.
↳ 0-shot, few-shot, chain of thought, prompt chaining.

L3 Instruc^n & Preference Tuning

- 0 shot learning performance is often poor → whatever be the model size. Fine tune the model on the instruc^n (1 approach is to reformat the available datasets into instruc^n sets)
  ↳ refer to FLAN dataset.
- We have to fine tune the model to align with user's intent. For this we need human labeller for a collec^n of prompts. Use it to fine-tune supervised fine tuning, the model using RLHF.

Fine-Tuning

SFT
task-specific full instruc^n tuning
preference tuning RLHF
memory efficient (PEFT, Quantiza^n)

Prompt
0 shot, few shot, chain of thought, prompt chaining, meta prompting.

- till now, we have trained the model now we need to fine tune it (HF will help you → peft, trl, SFT, bits and bytes, unsloth)

t4 - L6 → Demos