(LLM notes)
(Also see 106)

RAG (Retrival - Augmented generation)

- is a method in LLMs to access external knowlege (in Real time) rather than relying on only what they where trained on

it dose it by combining search (retrival) with language gen

- why: after training cut off date LLMs dont know anything and can hallucinate facts Rag solves this by
  - retriving relevent info from external knowlage base → via vector simmilarity → from DB's
  - generating answer using that retrived info (via LLM Eg GPT4)

A Rag chain is those structured sequence of steps a RAG pipeline uses Ex: input → Embedder → Retriver → Filter → LLM → Output

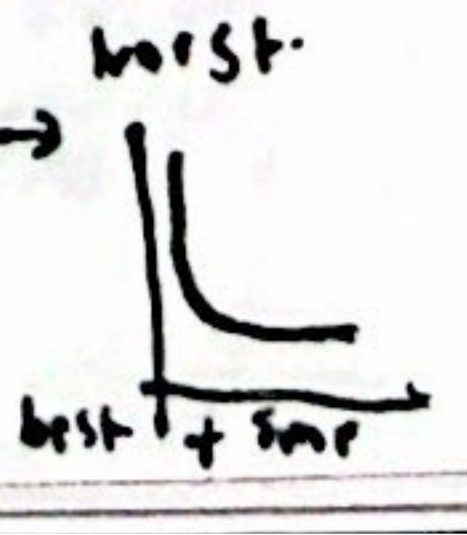
Fine tuning: involves using a pre trained model on specific dataset to adapt it to a particular task making it better for that task but limited to training knowlage, hence RAG

! (See page 104) ⇒ Scaling Laws for LLM

- describe how preformance improves as model size (prams), the data set and compute (FLOPs) change
- More compute → Bigger model + more data → Better preformance (up to a point)
- loss (perplexity) decreses as log-linearly as model size ↑, dataset size ↑ and compute ↑ until diminishing returns or overfitting
- key points/Laws
  - more params = Better capacity to learn complex patterns
  - more Data = Prevents overfitting, improves generalization
  - more training steps = more updates = more learning (up to a point)
  - more compute = enables longer training, more updates

Balance (Optimal ratios Between)
- model size • Data size • training compute
- if model too big for Dataset : overfit
- too much data for small model : underfit

Limits
- eventually you hit diminishing returns
- need better arcetechture (transformers, MoE)
- alignment + fine tuning (RLHF, RAG)

- This Limit of LLM ie getting better rapidly then slowing down → 

is the "Bending curve" and sol is mentioned above