(LLM notes)

RAG (Retrival - Augmented generation)

- is a method in LLMs to access external knowlage (in Real time) rather than relying on only what they where trained on it does it by combining search (retrival) with language gen

- why: after training cut off date LLMs don't know anything and can hallucinate facts Rag solves this by
  - retriving relevant info from external knowlage base → (via vector similarity 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 → Embeddor → Retriver → Filter → LLM → Output

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

Scaling Laws for LLM

- describe how preformance improves as model size (params), the dataset and compute (FLOPs) change
- More compute → Bigger model + more data → Better preformace (upto a point)
- loss (perplexity) decreases 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 compute = enables longer training, more updates
    - more training steps = more updates = more learning (upto a point)

Balance (Optimal ration Between)

- model size - Data size - training compute
- if model too big for Dataset = overfit
- too much data for small model = underfit

Limits

- eventualy you hit diminishing returns
- need better architechture (transformers, MoE)
- alignment & fine tuning (RLHF, RAG)

- This Limit of LLM ie getting better rapidly then slowing down → (worst → best + time)

it the "Bending curve" and sol is mentioned above