TLDW: State of GPT Summary Notes (Human Made 🙂):

- There are 4 Primary Steps in the GPT Assistant Training Pipeline
    - Pretraining
    - Supervised Finetuning
    - Reward Modeling
    - Reinforcement Learning

- Pretraining
    - 99% of this pipeline is spent in the Pretraining Stage
    - Dataset: large amount of publicly available data
        - Large quantity
        - Low quality
    - These datasets are often Tokenized (needed to be used as input for LLMs) into integers using the Byte Pair Encoding Algo
        - Vocab sizes range between 10K to 100K tokens
    - The inputs to the Transformer are (B,T), where B = batch size, and T = maximum context length
        - When concatenating datasets together, <|endoftext|> token is placed at the end of each document
        - Ex of (1,T) training example: Example document 2 <|endoftext|> Example document 3 <|endoftext|> Part of document 4
    - To train the Transformers during pretraining, next token prediction is employed, where the model can only use Tokens at time step T to 0 to predict the Token at T+1
    - Once pertaining is completed, will have a Base Model, like LLama
    - Base Models provide substantial benefits…
        - 1) For supervised tasks such as sentiment classification, base models fine tuned for the task of interest tend to perform better then models which were not pretrained
            - Reasoning: Through next token prediction during pretraining, the Transformer learned a lot about the structure of Text
        - 2) Base Models can be “tricked” into performing a certain task through effective prompting, where one example could be formatting a question as follows: Background info -> Example Q -> Example Ans: X -> Example Q -> Example Ans: X -> Actual Q -> Actual Ans: ?, where through the aim of “wanting” to complete the document it will answer the question (hopefully correctly)
            - The above is an example of few-shot prompting
    - However, even though I have highlighted above that Base Models can function as assistants when prompted appropriately, this is not very reliable. Thus, Supervised Finetuning is the next step employed in the GPT Assistant Training Pipeline

- Supervised Finetuning
    - Dataset: Human contractors are used to generate data in form of Prompt and Ideal Response
        - Low Quantity
        - High Quality

        - Inverse to Pretraining data
    - Similar to Pretraining, the Base Model is trained through next token prediction
    - Once Supervised Finetuning is completed, will have an SFT Model, but this is still not equivalent to ChatGPT

- Reward Modeling
    - Dataset: Provide inputs to SFT model and generate multiple completions, where humans will then rank the quality of these completions
    - We then train a Reward Model (video was vague, just highlighted Transformer NN, but other sources highlight using the SFT model), where when given the prompts and completions as input, it will learn to predict a high scalar reward for good responses and low scalar rewards for poor responses

- Reinforcement Learning
    - Dataset: Random selection of prompts generated by contractors
    - SFT model is provided prompt, in which it will generate a completion (using the RL Algo: Proximal Policy Optimization). This response and generation are passed to fixed reward model to generate reward score, which is used in the SFT model's loss function. This will in turn, promote higher probabilities for tokens pertaining to higher scoring completions and lower probabilities for tokens pertaining to lower scoring completions.
    - Generates RL Model, like ChatGPT
    - While RLHF models are superior assistants, these do tend to suffer from Mode Collapse, where it loses entropy (i.e more peaky token probability outputs) and in turn generates fewer variations of outputs. Base Models, in turn, have much more diversity

- Human Brain vs "LLM Brain"
    - Humans have an inner monologue which allow for them to think/reason and reflect on the task on hand, to varying degrees across different steps, whereas LLM's do not have this property
    - Human's know what they either don't know or are not good at (i.e 23450*76543), whereas LLM's do not have this property
    - However, unlike humans, LLM's do have large fact-based knowledge across numerous areas (stored in params) and large + perfect working memory (context window)
    - To allow for LLM's to perform better, need to make up for this cognitive difference between human and LLM brains. This is why Chain of Thought Prompting is important

- Chain of Thought
    - It is important to recognize that "Models need tokens to think", as often Transformers tend not to do too much reasoning per token
    - To aid this, breaking up your global tasks into multiple steps/stages and prompting it to have an internal monologue is important

    - Few-Shot COT Prompting is particularly effective, where if you give it a template of Q's and Answers, with work being shown to get the answers, the model should imitate the template when provided a following question
    - With GPT4, can even do effective Zero-Shot COT prompting by simply telling the model to think "step by step" after providing it a Q, forcing the model to use more tokens (and in turn reason more)

- Other techniques to improve performance of ChatGPT
    - Ensemble multiple completions
    - Ask the model to reflect on its outputs
    - Condition the model, via Ex. prompting "You are an expert on this topic", such that the Transformer (which is simply a token completer) generates outputs more inline with the training data which pertains to experts

- External information/tools which can be used to improve LLMs are…
    - Plugins, for tasks in which Transformers tend to perform poorly upon
        - Need to prompt the model: "You are not good at X, whenever need to do X, use tool Y, here is how to use tool Y…)
    - External information
        - I.E Retrieval-Augmented LLMs
            - Retrieve task-relevant information and incorporate into "working memory" context window
                - Will have embedding database composed of chunks of documents
                - In evaluation, retrieve related information with respect to query from database and incorporate in context

- What if you want to adapt a GPT specific to a certain domain, I.E Medical Domain, then need to finetune the LLM
    - Often you will conduct parameter efficient finetuning (PEFT), such as LORA
    - Most of the model weights are clamped, and will train small sparse pieces of the model
        - However, because the majority of weight are fixed, can use low precision inference when computing those parts (as we are not updating with gradient descent), which will improve efficiency
    - However, this is very difficult (especially the RLHF component) without expertise

- Final Recommendations for using ChatGPT
    - Use the latest GPT Model
    - Prompt the model with as much detail as possible (as if can only communicate with it once)
        - Try out the different prompting techniques highlighted earlier
        - Use Few-Shot Prompting, if possible
    - Incorporate external tools and information
    - If above fails, then consider creating own models
    - After satisfied, employ cost reducing measures (cheaper models, shorter prompts (pay by token) etc…)