AI&MLOps Module 3 to 4 Transition

Deepak Subramani
Assistant Professor
Dept. of Computational and Data Science
Indian Institute of Science Bengaluru

M04 Outline for Week 01

- Part 1: Decoder only GPT Model
  - What are GPT-class Generative Large Language Models
  - Generative AI Use Cases
  - Data preparation for GPT model training
  - GPT finetuning (Assignment)
- Part 2: LLMs and Interacting with them
  - Commercial and open source LLMs
  - What are the main issues in LLMs to be aware of?
  - Taxonomy of interaction with LLMs
  - Parameter Efficient Fine Tuning (LoRA, QLoRA)

M04 Outline for Week 02

- Part 1:
  - Prompting Strategies – ZSL, FSL, CoT, ReACT, DSP
- Part 2:
  - Instruction Tuning
- Part 3:
  - Orchestration → Agentic AI
  - Retrieval Augmented Generation
- Part 4:
  - LLM Guardrails
  - LLM Agents

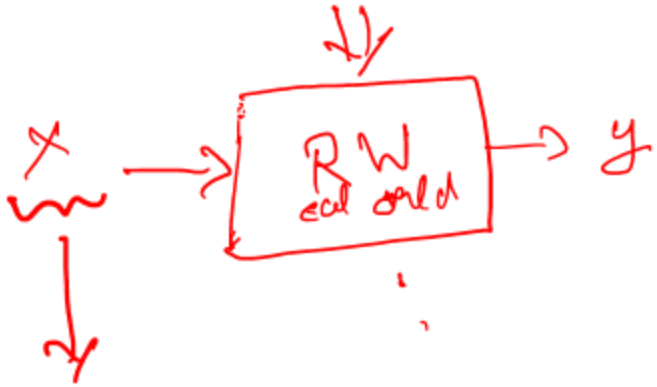
Deepak Subramani, deepakns@iisc.ac.in

M04 Outline for Week 03

- Part 1:
  - Deep learning as a representation learning system
  - Autoencoders for pre-training new situations
- Part 2:
  - Modern GenAI Image Pipelines
  - CLIP
  - Stable Diffusion
- Part 3: (May Know)
  - GANs – Generative Adversarial Networks
  - Variational Auto Encoders
  - Resource for Math of Diffusion Models (Link Shared in Part 2)

Predictive AI and Generative AI

- Predictive AI
  - Input: Any of the data modality
  - Output: Continuous or Categorical
- Generative AI
  - Input: Any of the data modality
  - Output: Text, Image, Video, Audio



Diffusion model Transformer

Stable Diffusion

Modality

Model

How does a transformer encoder work?

Explain the inner working of a neural text classification system.

Multi modal LLMs

→ multi modal Large Language Model

Multi modal multi model 4o

Generative AI

- Text to Text
- Text to Image/Video
- Image/Video to Text
- Image/Video to Image/Video
- Text to Audio
- Audio to Text
- Text/Image to Code

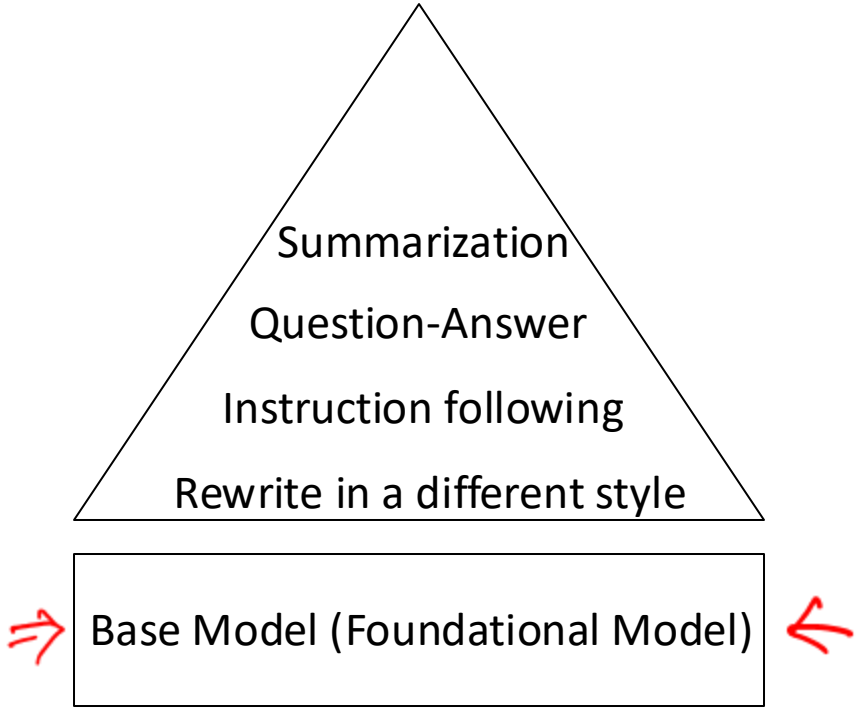
- Input is the “Prompt”; Model is a Large Language/Vision Model; Output is Image/Video/Text/Speech

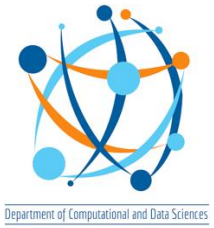
Deepak Subramani, deepakns@iisc.ac.in

Foundational Model

Build to Print

- Large-scale AI model trained on vast amounts of diverse data
- Serves as a base for multiple downstream tasks and applications
- Key characteristics:
  - Broad knowledge and capabilities
  - Prompt engineering to make it perform tasks
  - Retrieval Augmented Generation for tapping into specific data
  - Adaptable through fine-tuning
  - Generalize to new tasks with minimal additional training
- **Examples**: GPT, BERT, T5





Generative AI Use Cases

- **Healthcare Assistance** – Offering support in areas like patient interaction, medical documentation, and even as assistive tools for diagnosis and treatment planning, though they don't replace professional advice.
- **Personal Assistants** – Managing schedules, setting reminders, answering questions, and even helping with email management and other administrative tasks.
- **Legal and Compliance Assistance** – Assisting in legal research, document review, and drafting legal documents (without replacing professional legal advice).
- **Accessibility Tools** – Enhancing accessibility through tools like voice-to-text conversion, reading assistance, and simplifying complex text.
- **Interactive Entertainment** – In gaming and interactive storytelling, creating dynamic narratives, character dialogue, and responsive storytelling elements.
- **Marketing and Customer Insights** – Analyzing customer feedback, conducting sentiment analysis, and generating marketing content, providing valuable insights into consumer behavior.
- **Social Media Management** – Managing social media content, from generating posts to analyzing trends and engaging with audiences.
- **Human Resources Management** – Aiding in resume screening, answering employee queries, and even in training and development activities.

Deepak Subramani, deepakns@iisc.ac.in

Generative AI Use Cases

- **Customer Service and Support** – Providing customer support, handling inquiries, resolving issues, and offering information 24/7.
- **Content Creation and Copywriting** – Generating creative content, such as articles, blogs, scripts, and advertising copy.
- **Language Translation and Localization** – Translation services for various content types, aiding in bridging language barriers and localizing content for different regions.
- **Education and Tutoring** – Functioning as personalized tutors, providing explanations, answering questions, and assisting with learning materials in a wide range of subjects.
- **Programming and Code Generation** – Writing, reviewing, and debugging code, thereby speeding up the development process and helping in learning programming languages.
- **Research and Data Analysis** – Sifting through large volumes of text, summarizing information, and extracting relevant data, which is invaluable for research and analysis.

Text 2 Table Table 2 Text Text 2 SQL SQL 2 Table

Latest Developments

- Anthropic
  - Claude 3.5 Sonnet
- Microsoft-OpenAI integration
  - Bing search
  - PowerBI with ChatGPT
- Generative Image
  - Photoshop
  - MidJourney
- Generative Videos
  - Sora
- LLMOps pipeline
  - LangChain/LlamaIndex + OpenAI/Antrhopic/Llama

Language Model

- Any model that can predict the probability of the next token in a sequence of text input (converted to embeddings) is called a Language Model
- LM captures the latent space of language: its statistical structure
- Large Language Models are trained on large text corpora (trillions of tokens) and have billions of parameters
- They have emergent abilities
  - Can do tasks for which it is not explicitly trained
  - Today, we don't take a chance and make it learn to follow instructions
- Finally, LLMs would all be a sophisticated lookup table!

Language Modeling Approaches

- Masked Language Modeling
  - Tokens in a document are randomly masked
  - Neural Models are trained to predict the masked token correctly
  - This is a fill-in-the-blank task
  - Example:
    - The cat sits on the mat.
    - The [MASK] sits on the mat.
    - The model's task is to predict "cat" based on the context
- Sentence Completion Modeling (Next token prediction)
  - Model is set up in an autoregressive mode
  - At each inference step, the model predicts the next token (from the vocabulary as a probability distribution)
  - (k+1)st token is predicted with (prompt+predicted k tokens) as input
  - (k+2)nd token is predicted with (prompt+predicted k+1 tokens) as input

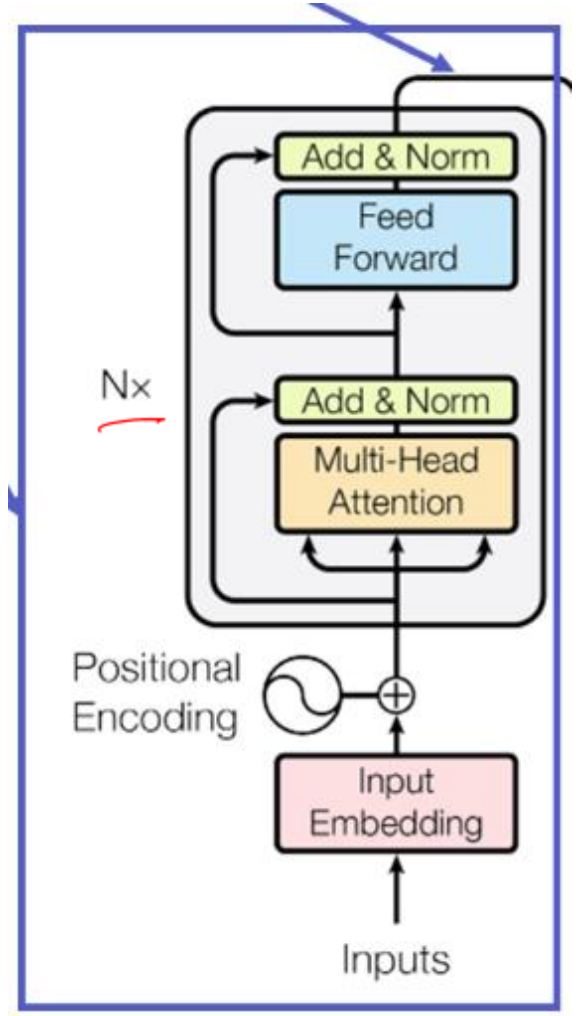
Transformers

- Transformer Encoder
  - Converts a sequence of words to a vector representation
  - This vector representation can be used for text-understanding tasks
  - Trained using fill-in-the-blanks tasks - MLM
- Transformer Decoder
  - Uses the context of the sequence of words so far (sometimes with an additional context from encoder or retrieval) to predict next token in the sequence
  - Trained using next-token-prediction tasks

Transformer Encoder

Steps in a Transformer Encoder

1. Tokenize (append special tokens – [CLS])
2. Get Encoded sequence added with position
3. Multi-headed attention
   - Converts encoded sequence to context aware representation (still a sequence)
4. Residual and layer normalization
5. Dense Layers for further representation learning
6. Encoder block outputs a encoded representation
7. Use the representation of [CLS] token to perform text understanding tasks



Transformer Decoder

Steps in a Transformer Decoder

1. Tokenize the prompt
2. Get Encoded sequence added with position
3. Masked Multi-headed attention
   - Masking makes the attention attend only to tokens to the left
4. Residual and layer normalization
5. Dense Layers for further representation learning
6. Decoder block outputs sequence of representations
7. Use the representation of last token to generate the next token
8. Repeat by including the generated token as part of the prompt