HCLTech | Supercharging Progress™

Cloud Native & AI Labs

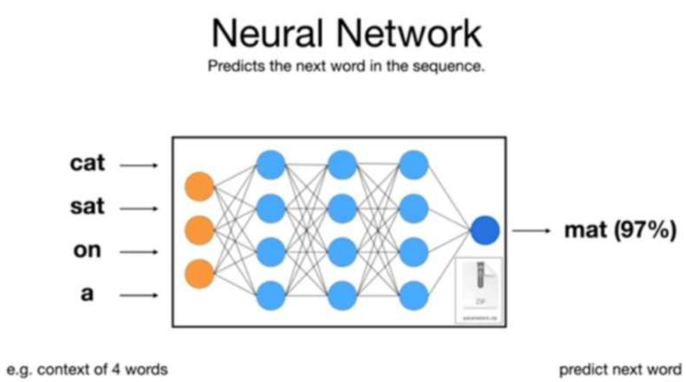
LLMs and Generative AI

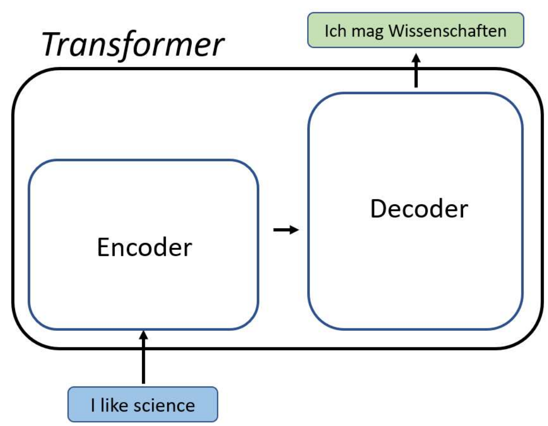
HCL

Large Language Models

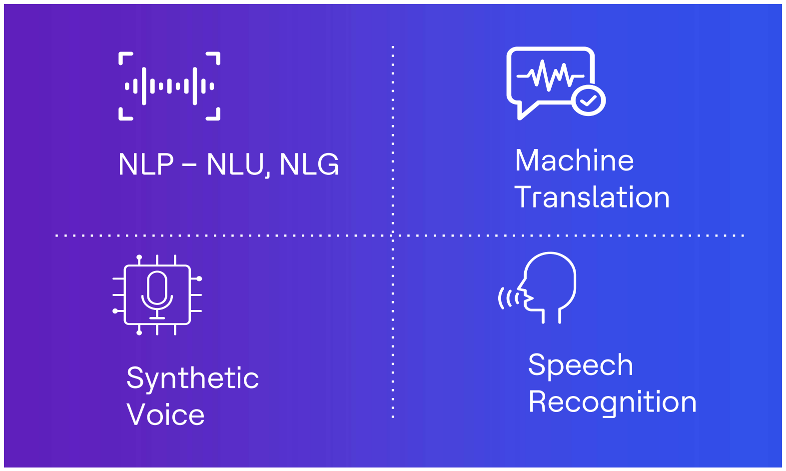
LLMs are trained to solve language problems like…

- Text Classification
- Question Answering
- Document Summarization
- Text Generation

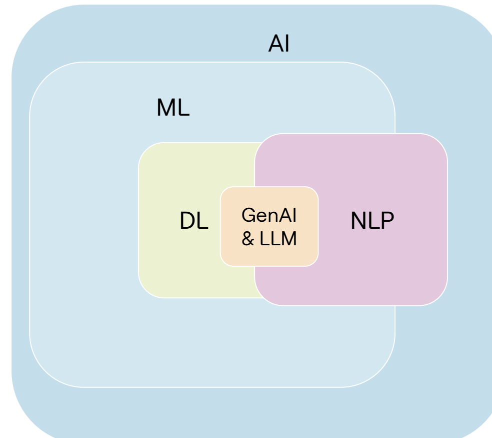




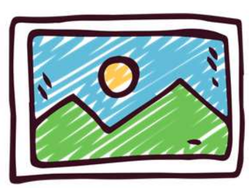
Generative AI



Artificial Intelligence

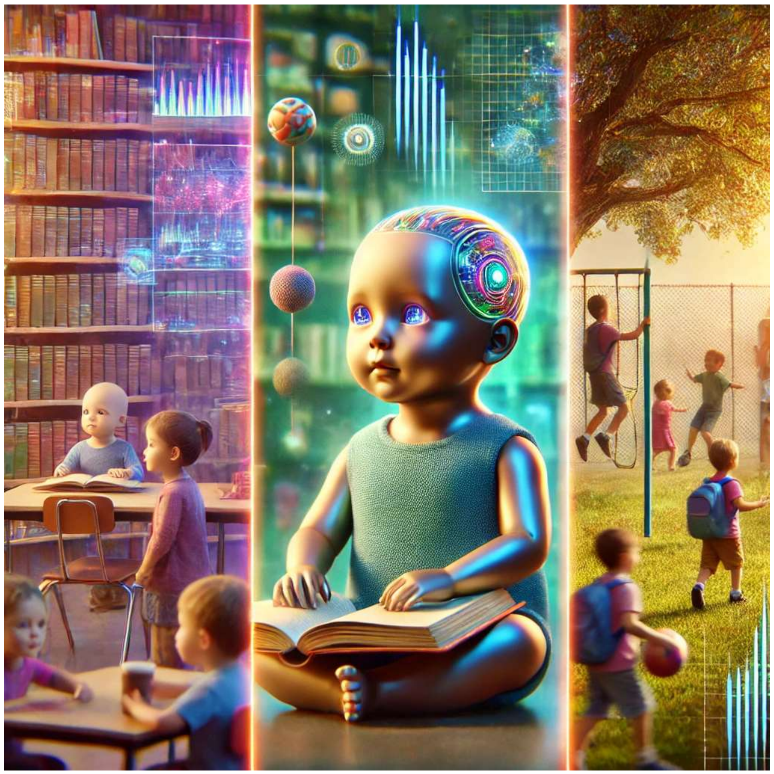


Domains of Generative AI



AI & Cloud Native Labs

The Philosophy of Learning



1. The world is chaotic, and our brains are optimized to learn and adapt to chaos.

2. Imperative → Declarative Paradigms → AI/Machine Learning

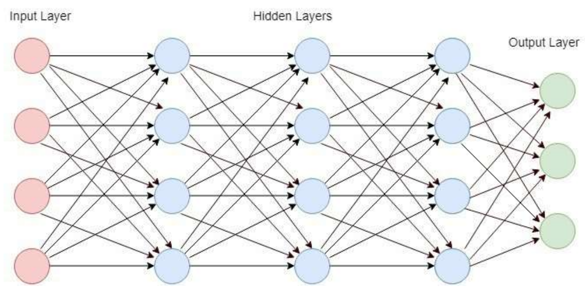
3. Types of Learning:
- Supervised
- Unsupervised
- Reinforcement
- Self-supervised

1. Word ordering
2. Image jigsaw
3. Denoising (Diffusion)
4. Game of compression

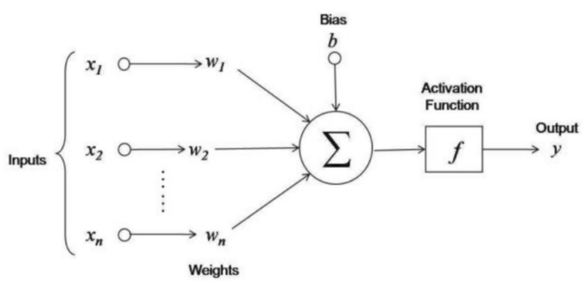
We are looking at new concepts by looking at the relationships between different concepts. We also reason in Language.

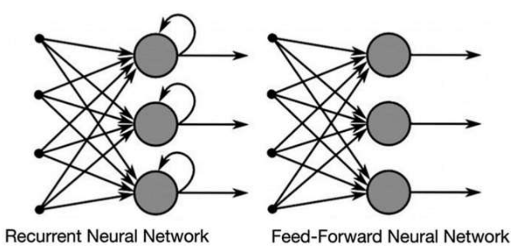
Cat kitten Rabbit
kit
Dog Puppy

Neural Networks

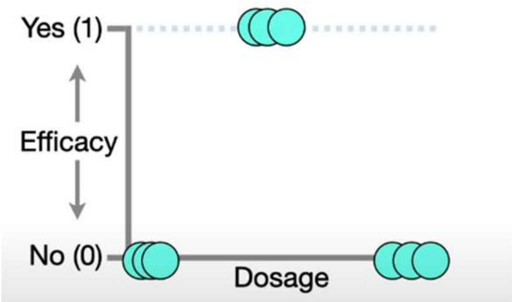


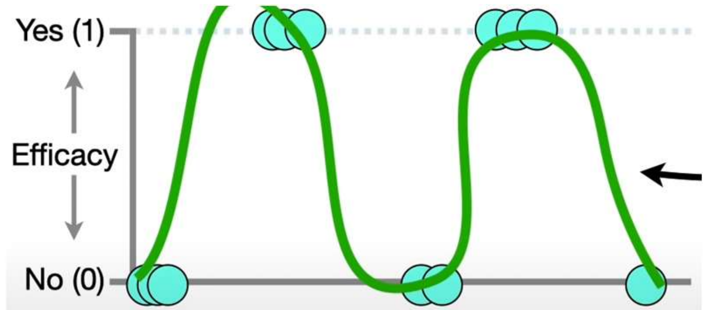
- Neurons - Fundamental unit of processing in a neural network.
- Neural network – a network of interconnected nodes or neurons in a layered structure that resembles the human brain. Used for machine learning process called deep learning
- 3 layers in a neural network – Input, Hidden, Output
- Types of neural network:
    - Feedforward Neural Networks
    - Recurrent Neural Networks
    - Convolutional neural networks
    - Generative Adversarial Networks
    - Etc.



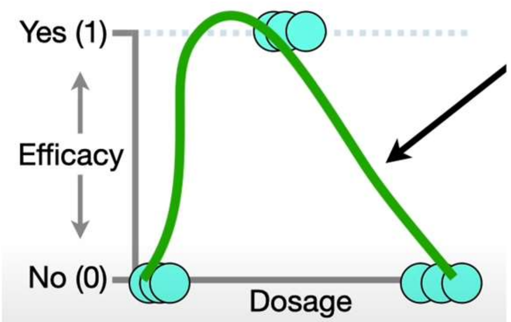


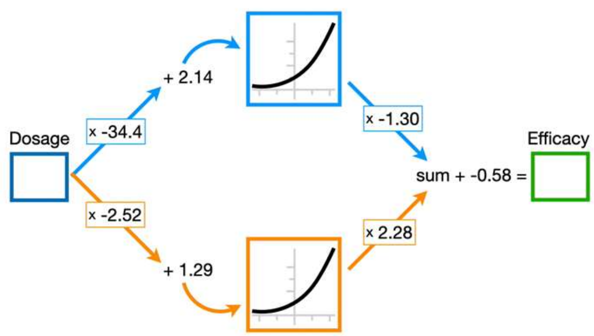
Neural Networks



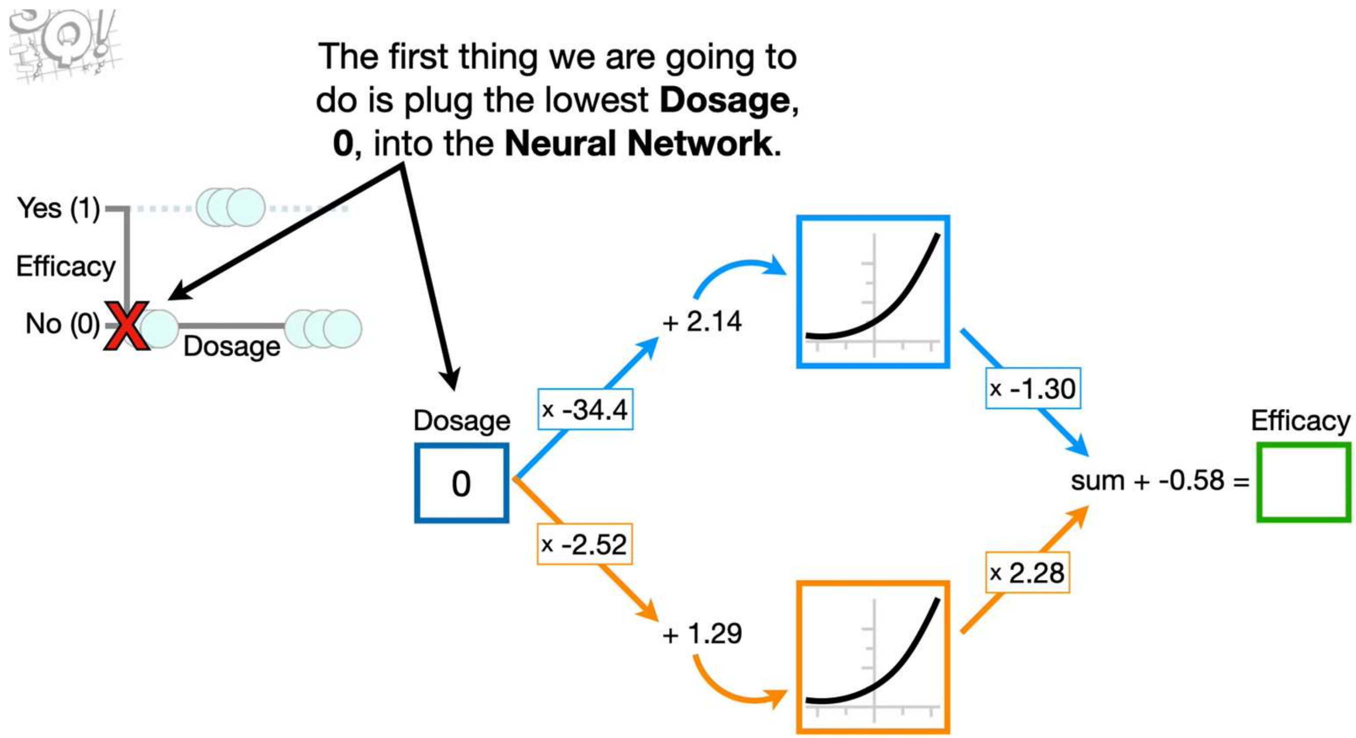


https://deepnote.com/app/martin-alvarez-59be/Neural-Networks-1c9e6060-64bc-412e-9539-aeb20ef76214

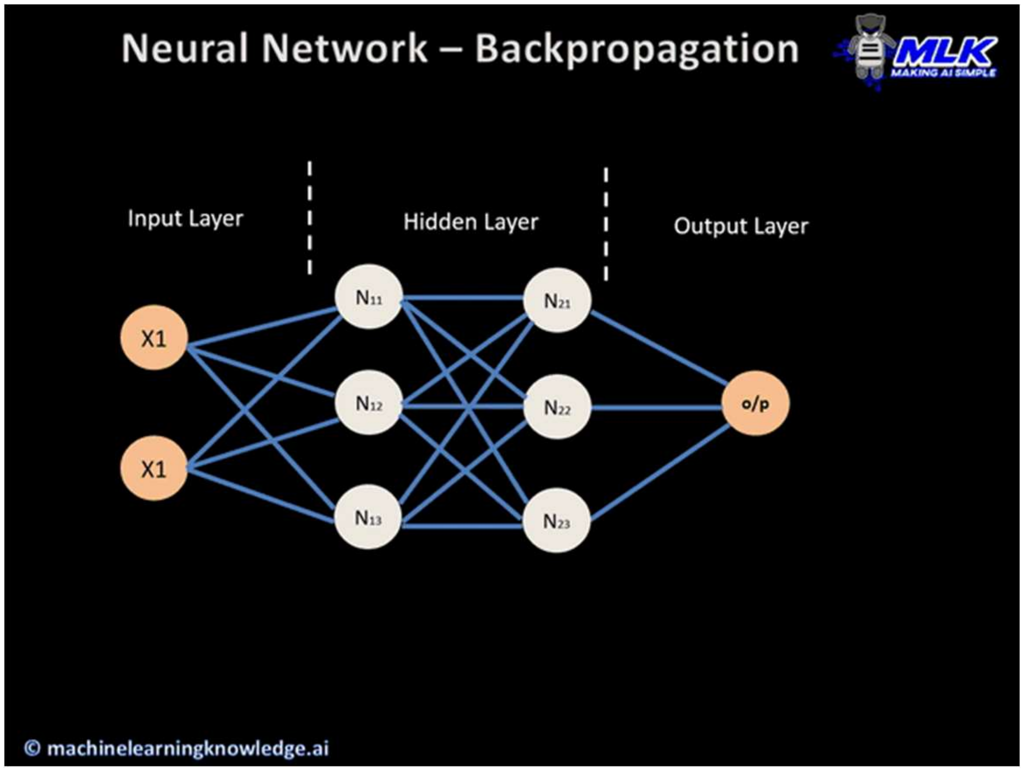


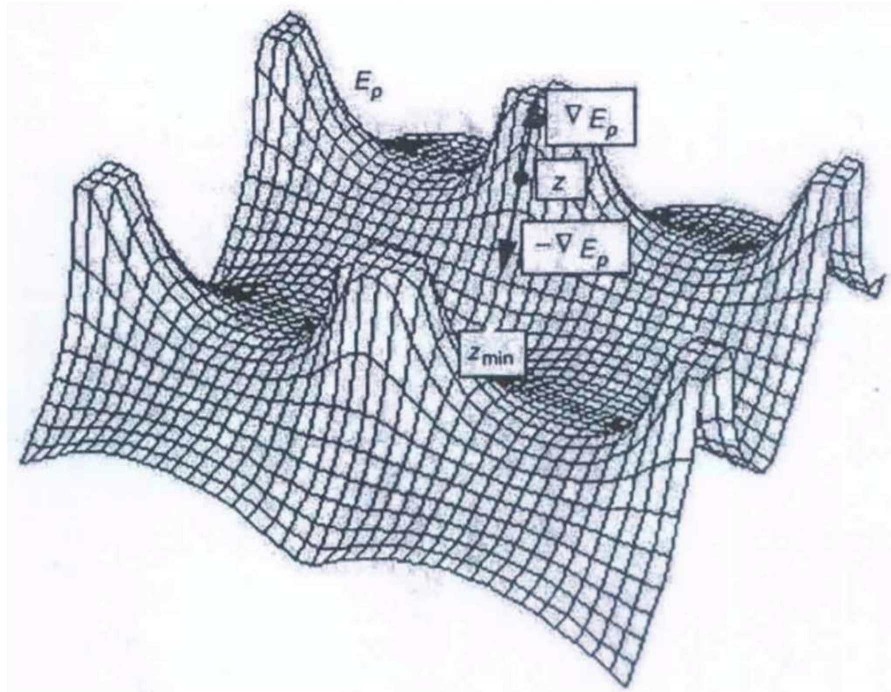


Neural Networks



Backpropagation

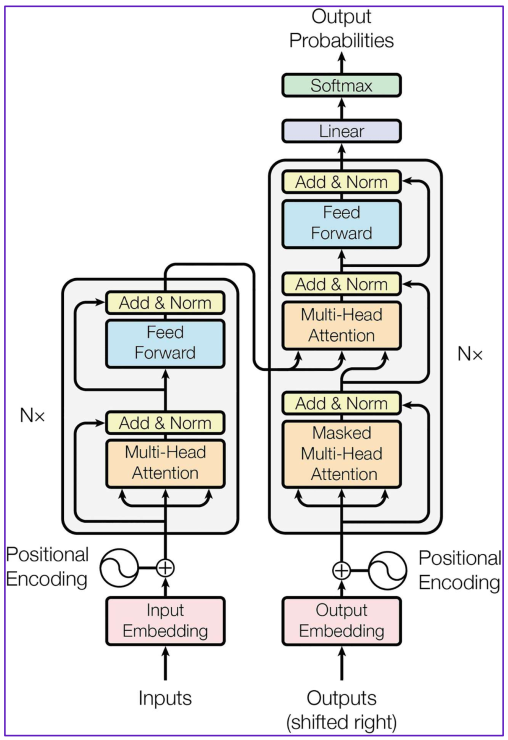




Transformers

Seq2Seq Neural Network

Transformers



- **Self-Attention Mechanism** - allows the model to assign weights to different words in a sentence based on their importance and then compute a weighted sum of values
- **Multi-Head Attention** - To capture the relationships between the words in a sentence, the transformer uses multiple sets of self-attention mechanisms, known as heads. Each head learns different aspects of the relationships, and their results are concatenated and linearly transformed.
- **Positional Encoding** - Unlike RNNs, transformers don't have an inherent sense of order due to their parallel processing behavior. Positional encodings are added to the input embeddings to provide information about the position of each token in the sequence.
- **Encoder-Decoder Architecture** - The original transformer architecture consists of an encoder and a decoder. In machine translation, the encoder processes the source language sentence, and the decoder generates the target language translation.
- **Pretraining and Fine-Tuning** - Transformers are often pretrained on large set of data using self-supervised tasks. After pretraining, the models can be fine-tuned for specific tasks, adapting their knowledge to the target application.

{ LLM is two files = ① code ② parameters }

for Llama 2   70B parameters x 2 Bytes   140 GB ← NN weights

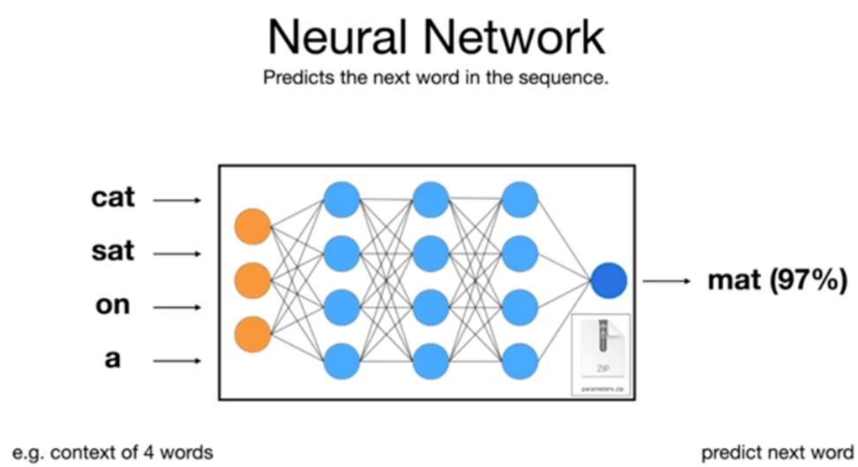
140 GB is self-contained ()

Training (compression) ~ 10TB ⟶ $\frac{\text{6000 GPUs}}{\text{12 days}}$ ⟶ ~140GB

→ Llama 3

→ Phi3

* Lossy compression



LLM

$\boxed{\text{parameters}}$ 140 GB

$\boxed{\text{run.c}}$ 500 lines

llama-2-70b

~10 TB = 6000 GPUs for 12 days = ~$2M → 140 GB (Lossy compression)

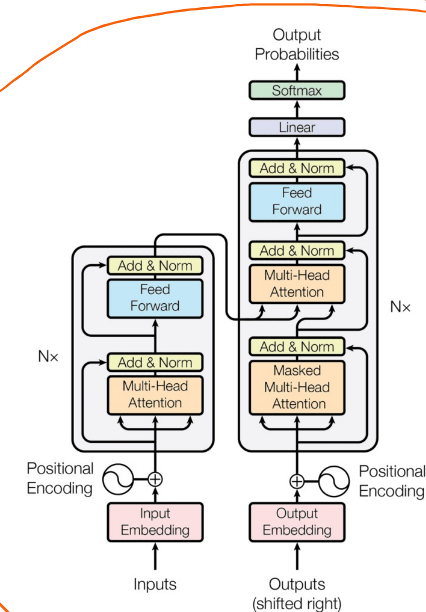
Mohsen is a person ← transformer architecture dreams Internet
Probablity

it is an odd way of compression

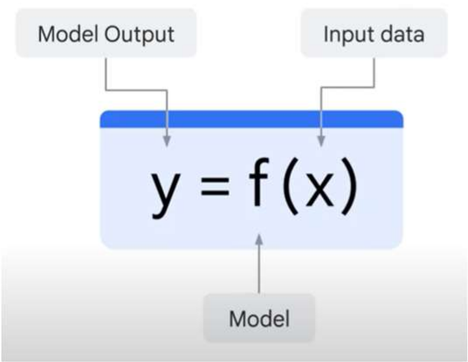
① Base model
Transformer
PreTraining
Tons of Internet
Every Year

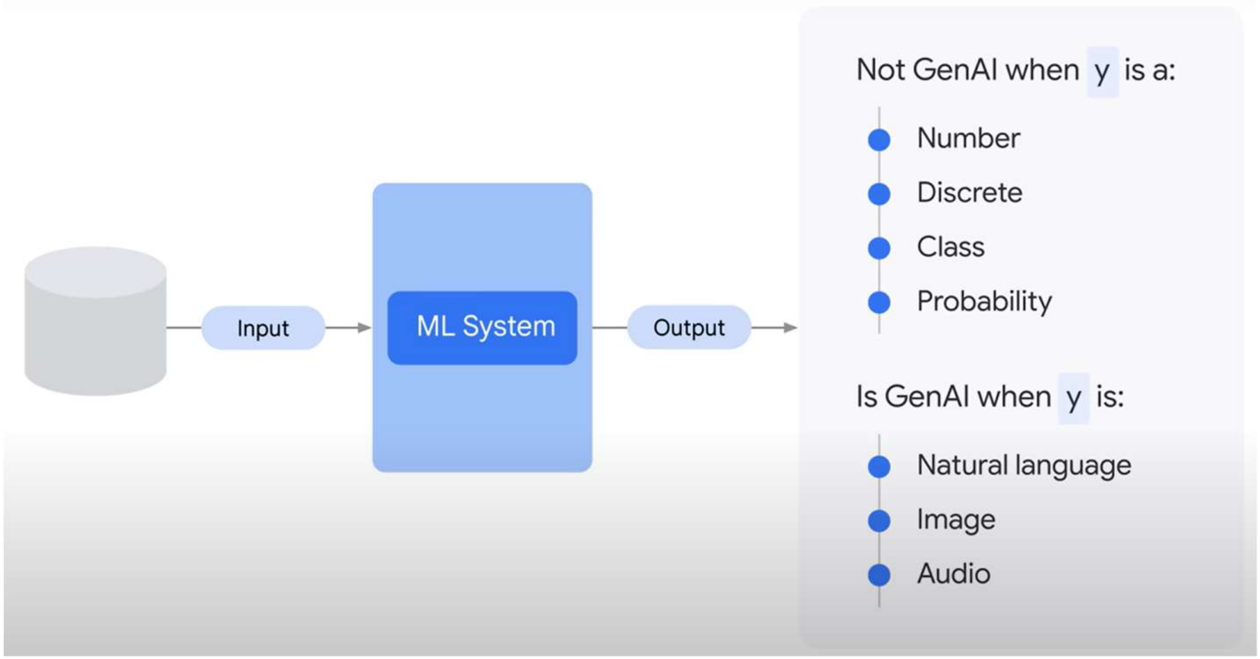
② assistant Model
1000k answers and convo
"Fine Tunining"
helpful assistant
Every week

③ Comparison stage
RLHF / Classification



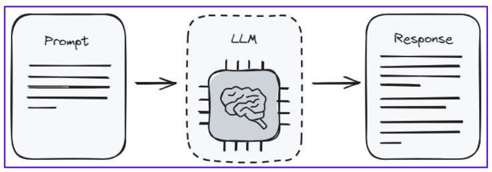
Gen AI vs Not Gen AI



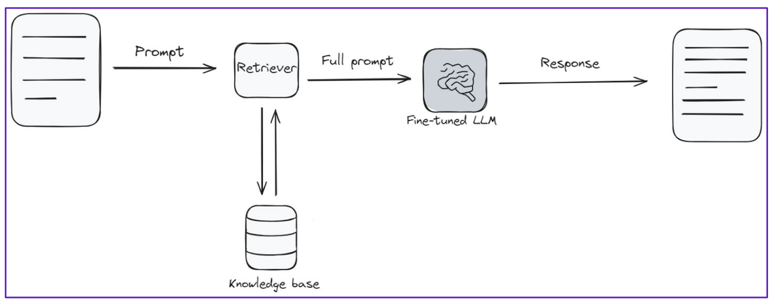


Generative AI – Fetching Data from LLMs

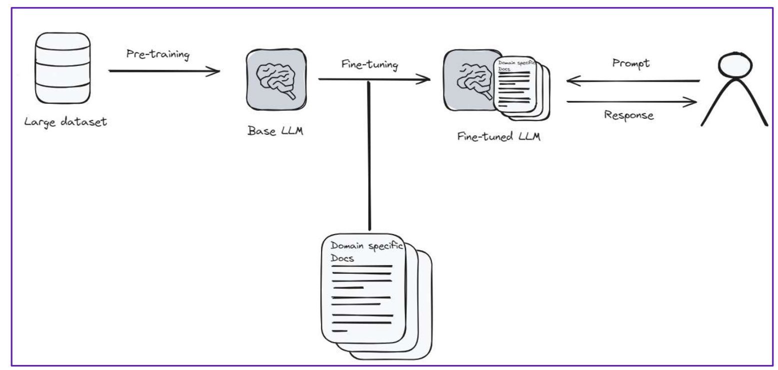
Prompting → Retrieval Augmented Generation → Fine Tuning → Model Training



Prompt Engineering
- Easy to use
- Cost effective
- Flexible
- Inconsistent
- Limited customization
- Dependency on model's knowledge



RAG
- Dynamic information
- Cost Balance
- Contextual data
- Complexity
- Resource intensive
- Data dependency
- Performance dependency



Fine Tuning
- Allows for extensive customization
- Improved Accuracy
- Adaptability
- Expensive
- Technical skills
- Data dependency

Gen AI Application Journey

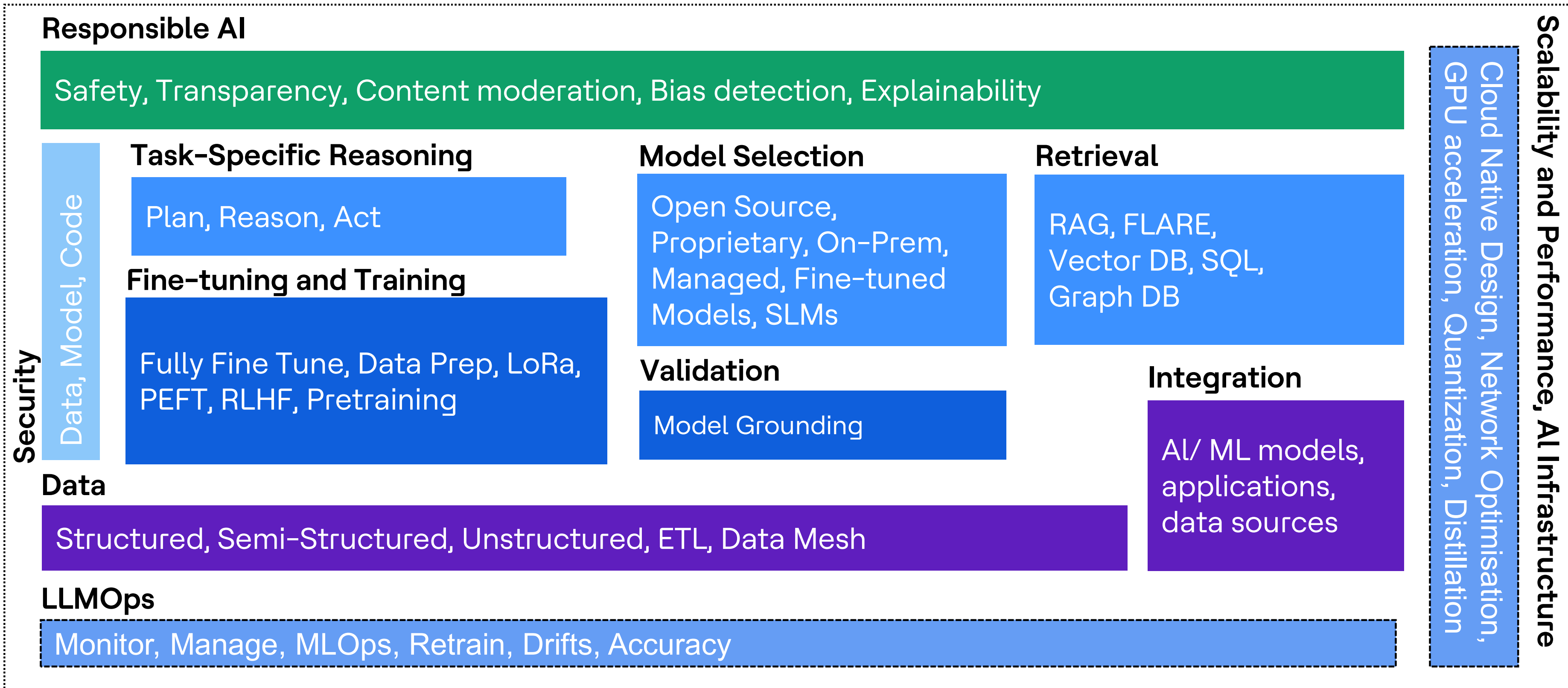
Level 5	Multi-Agents, LLMOps, Adv. Orchestration, LLM Mesh, SLM
Level 4	Fine-tunning, RAFT, Hybrid AI
Level 3	Agentic RAG (ReAct), Grounding & Evaluation
Level 2	RAG (Retrieval Augmented Generation)
Level 1	Prompt Engineering and Model Access
Level 0	Data Access and Prep

Security and Responsible AI

ROI

HCLTech

Common Reference Architecture



A common reference architecture underpins consistency and efficiency across Gen AI solution development, serving as a blueprint for scalable and robust implementation. Whilst following a common logical architecture, it will be specialized for each vendor platform.

HCLTech