

Technical Foundations Of Gen AI And Analytical Thinking

Dr. Avinash Kumar Singh

Robotics and Artificial Intelligence Training Academy

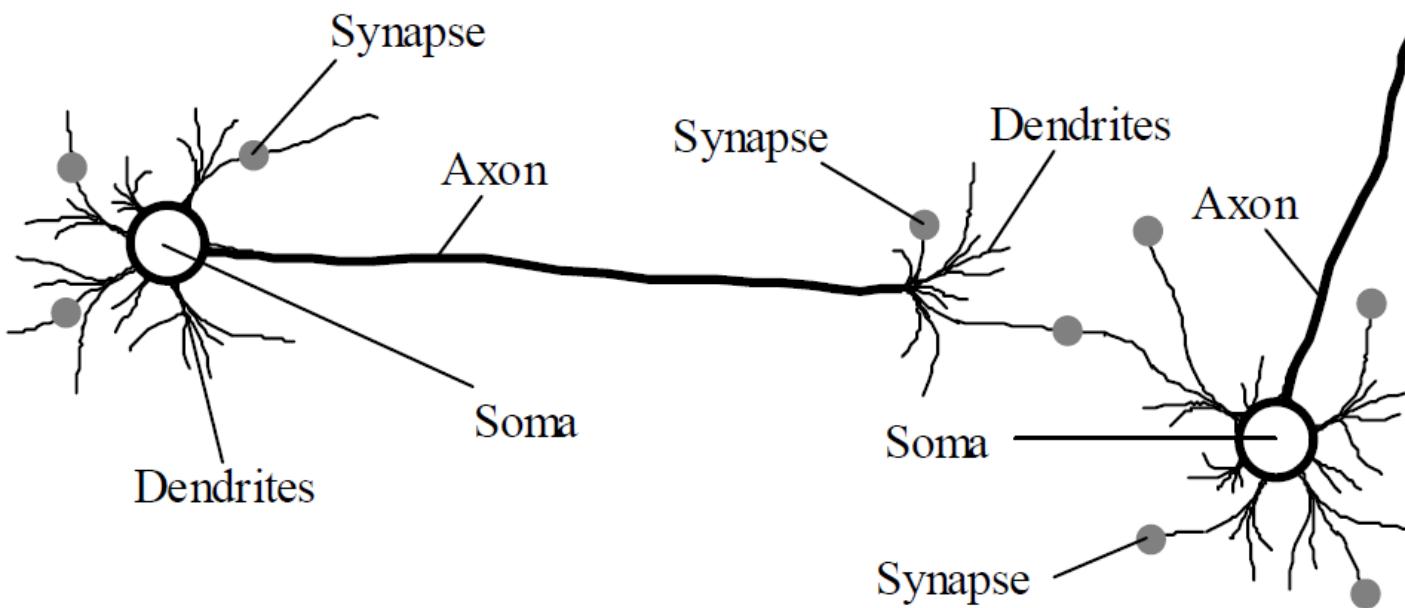


Discussion Points

- Artificial Neural Network
- Examples
- Large Language Models
 - Journey
 - How they work, trained
 - Evaluation
- Retrieval Augmented Generation
- Generative AI VS AI Agent VS Agentic AI

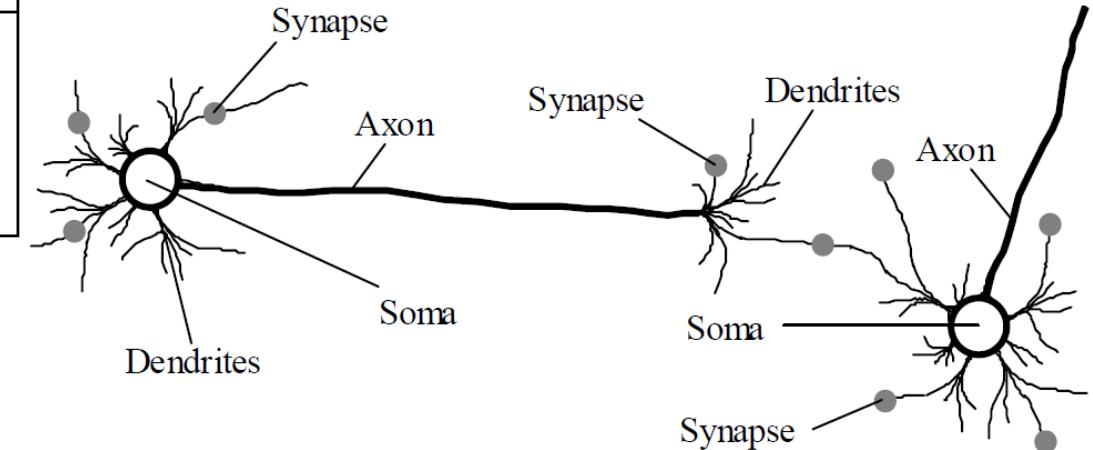
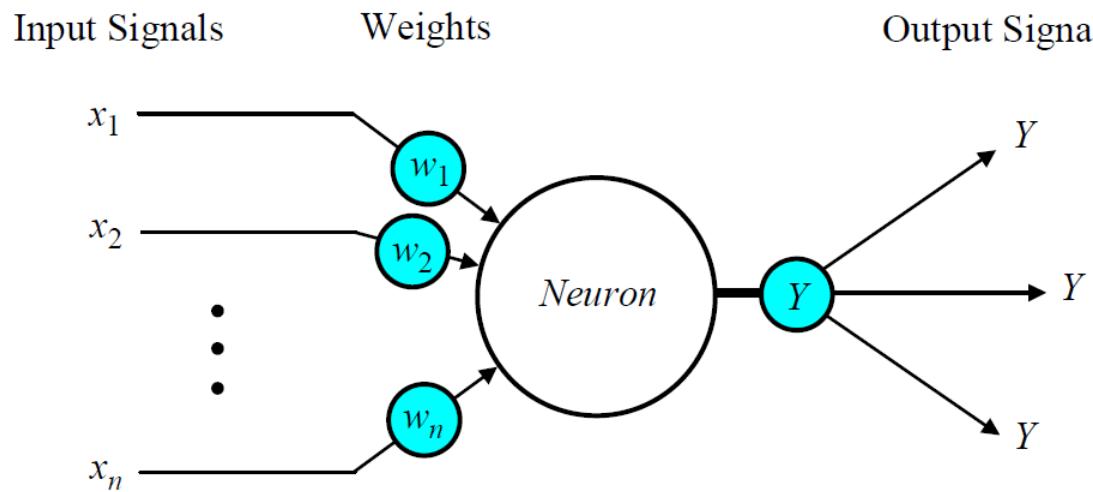
How Brain Works

- The human brain incorporates nearly 10 billion neurons and 60 trillion connections, synapses, between them.
- A neuron consists of a cell body, soma, a number of fibers called dendrites, and a single long fiber called the axon.

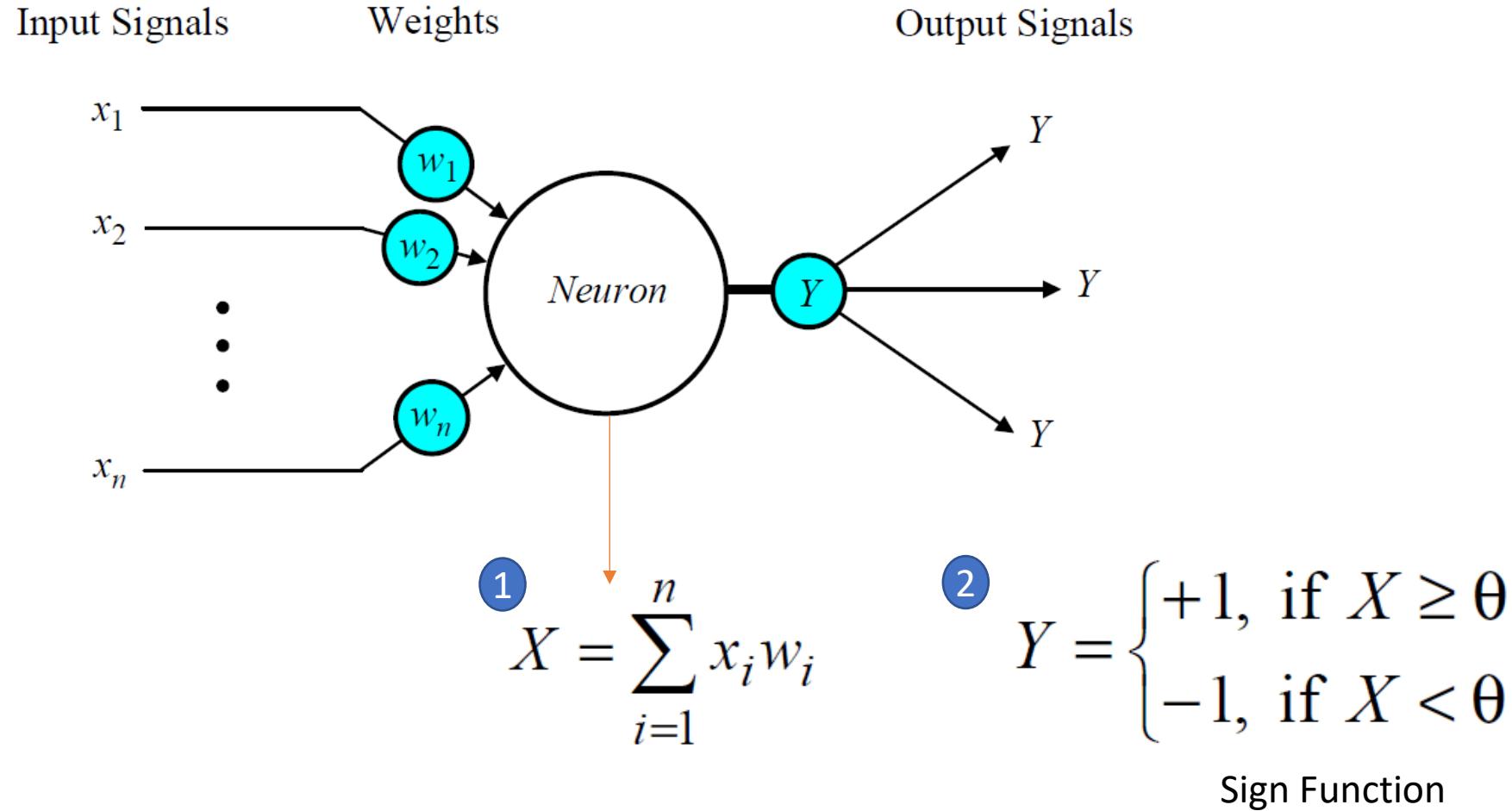


Similarity between brain and ANN

<i>Biological Neural Network</i>	<i>Artificial Neural Network</i>
Soma	Neuron
Dendrite	Input
Axon	Output
Synapse	Weight

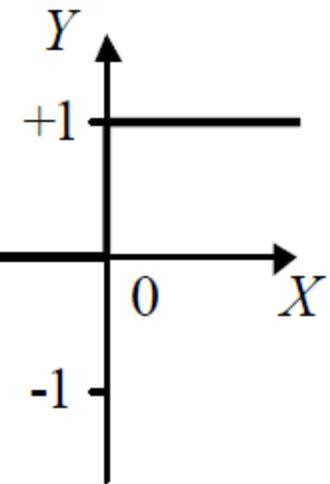


Perceptron Learning

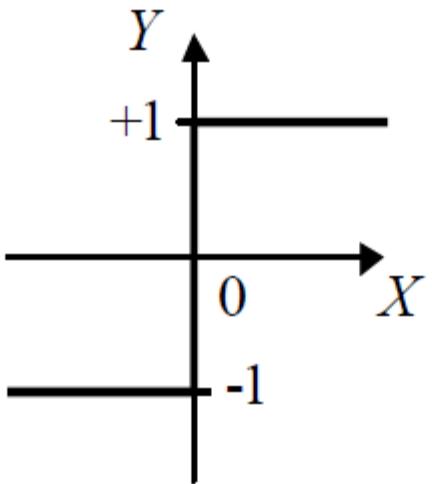


Activation Functions

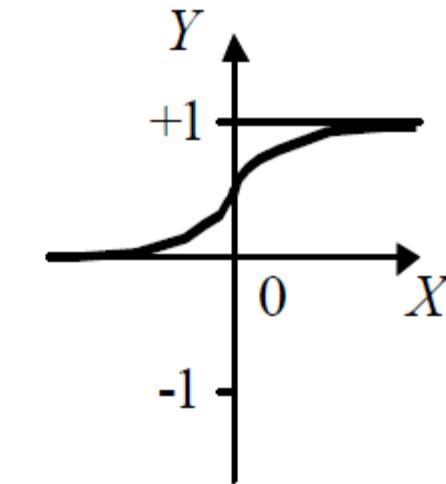
Step function



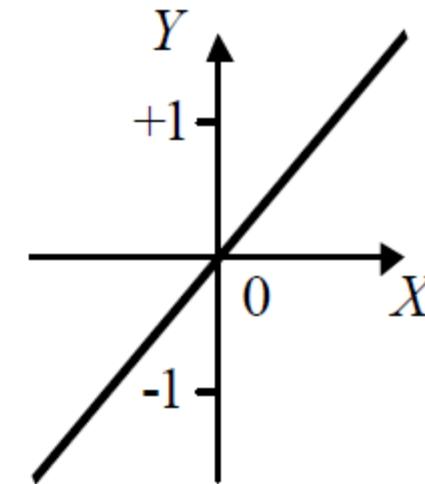
Sign function



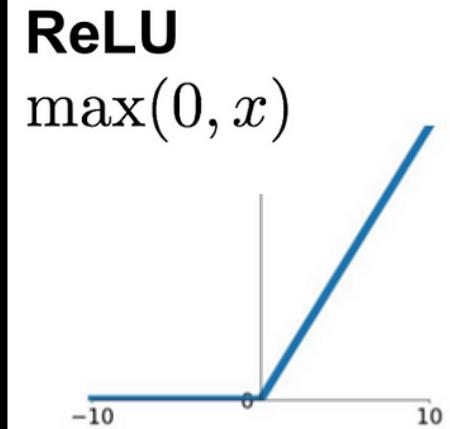
Sigmoid function



Linear function



Rectified Linear Unit (ReLU)



$$Y^{step} = \begin{cases} 1, & \text{if } X \geq 0 \\ 0, & \text{if } X < 0 \end{cases}$$

$$Y^{sign} = \begin{cases} +1, & \text{if } X \geq 0 \\ -1, & \text{if } X < 0 \end{cases}$$

$$Y^{sigmoid} = \frac{1}{1+e^{-X}}$$

$$Y^{linear} = X$$

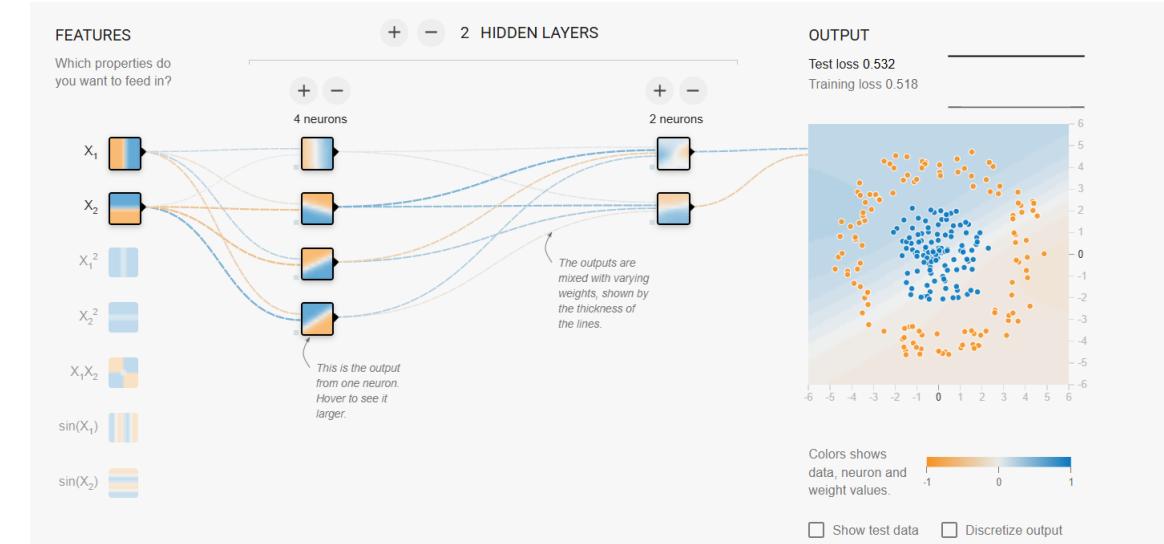
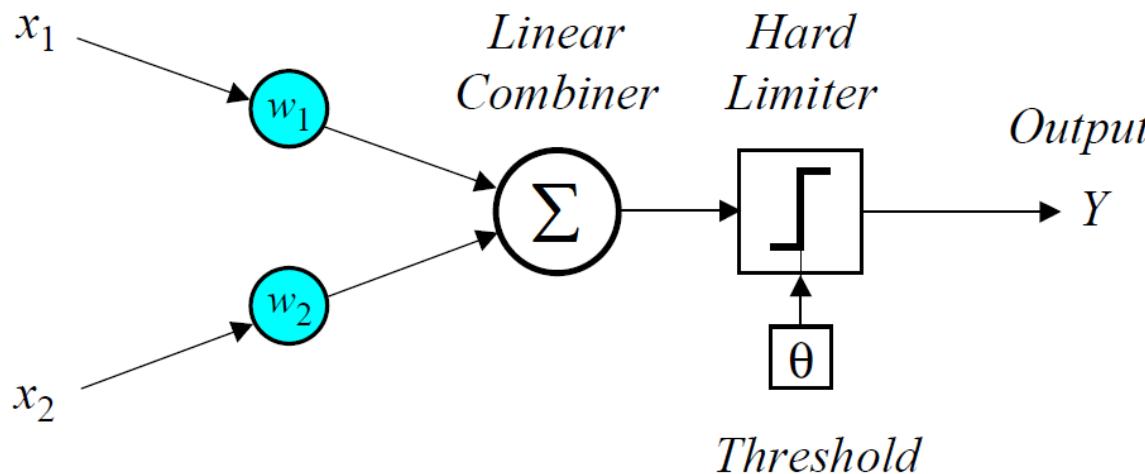
$$R(x) = \begin{cases} x & \text{if } x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

How a perceptron learns

- In 1958, Frank Rosenblatt introduced a training algorithm that provided the first procedure for training a simple ANN: a perceptron, inspired by

McCulloch and Pitts neuron model

Inputs



How a perceptron learns

Inputs

$$x_1 \rightarrow w_1$$
$$x_2 \rightarrow w_2$$

Linear
Combiner

$$\sum_{i=1}^n x_i w_i$$

Threshold

Hard
Limiter

$$\text{Step}(\sum_{i=1}^n x_i w_i - \theta)$$

Output
 Y

AND Problem

x_1	x_2	y_d	w_1	w_2	y_a
0	0	0	0.3	-0.1	
0	1	0			
1	0	0			
1	1	1			

$$1 X = \sum_{i=1}^n x_i w_i$$

$$2 \sum_{i=1}^n x_i w_i - \theta = 0$$

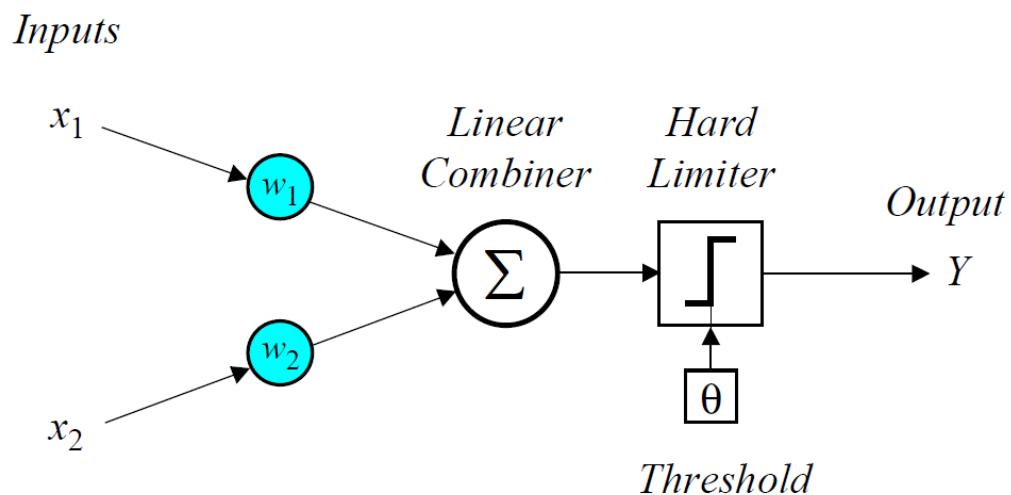
$$\text{Step}\left(\sum_{i=1}^n x_i w_i - \theta = 0 \right)$$

Threshold: $\theta = 0.2$; learning rate : $\alpha = 0.1$; activation function : step

How a perceptron learns

x_1	x_2	y_d	w_1	w_2	y_a	Error	w_1^{new}	w_2^{new}
0	0	0	0.3	-0.1				
0	1	0						
1	0	0						
1	1	1						

Example



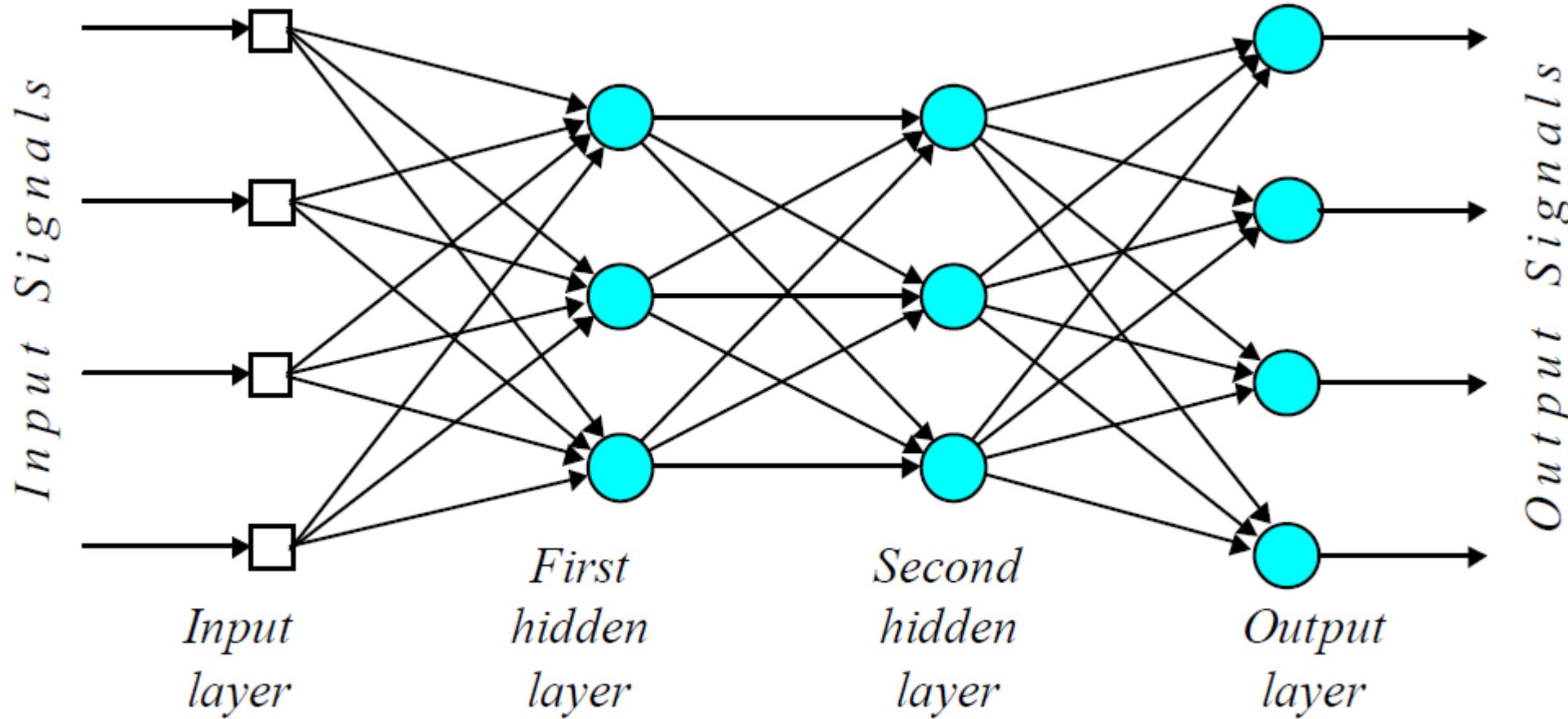
Weight Updation

$$w_i(p+1) = w_i(p) + \Delta w_i(p)$$

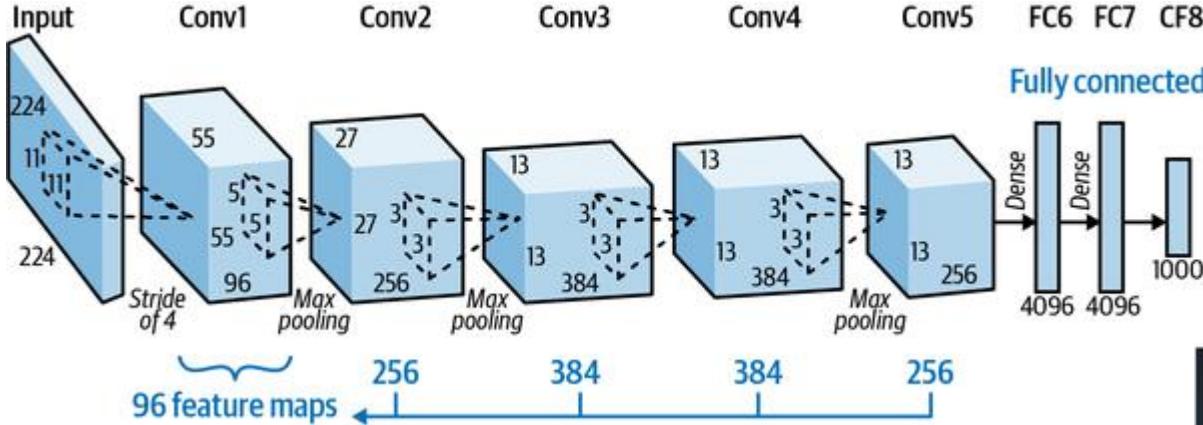
$$\Delta w_i(p) = \alpha \cdot x_i(p) \cdot e(p)$$

Threshold: $\theta = 0.2$; learning rate : $\alpha = 0.1$; activation function : step

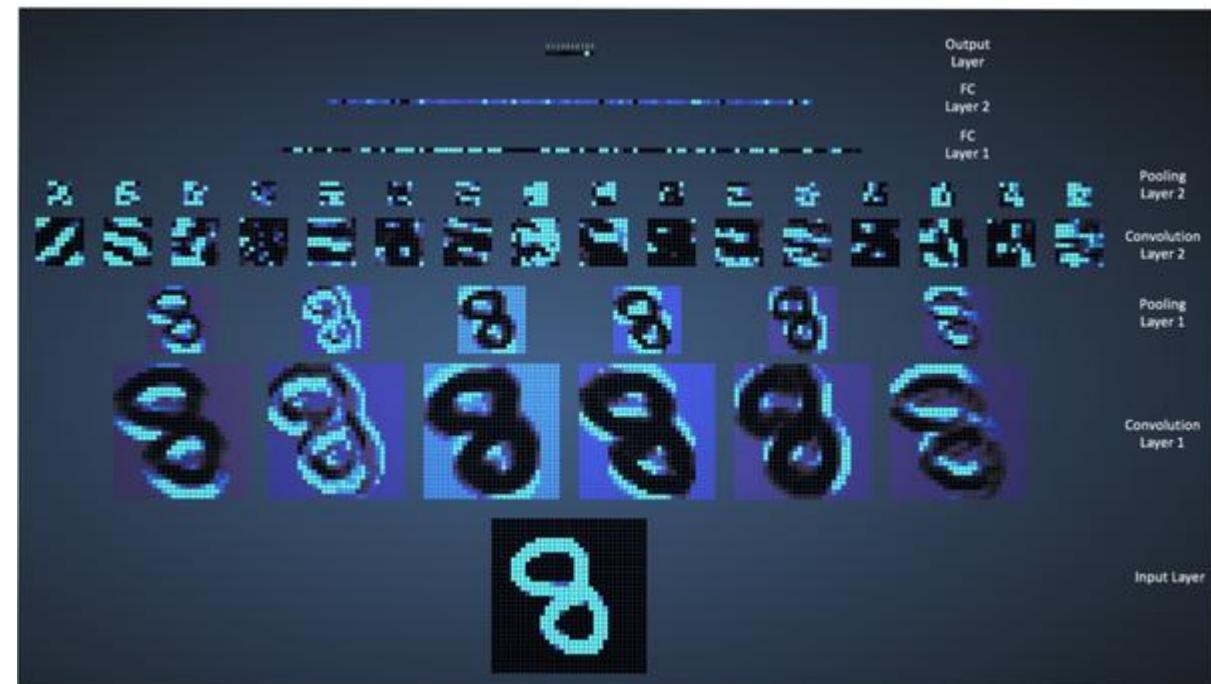
Multi Layer Perceptron



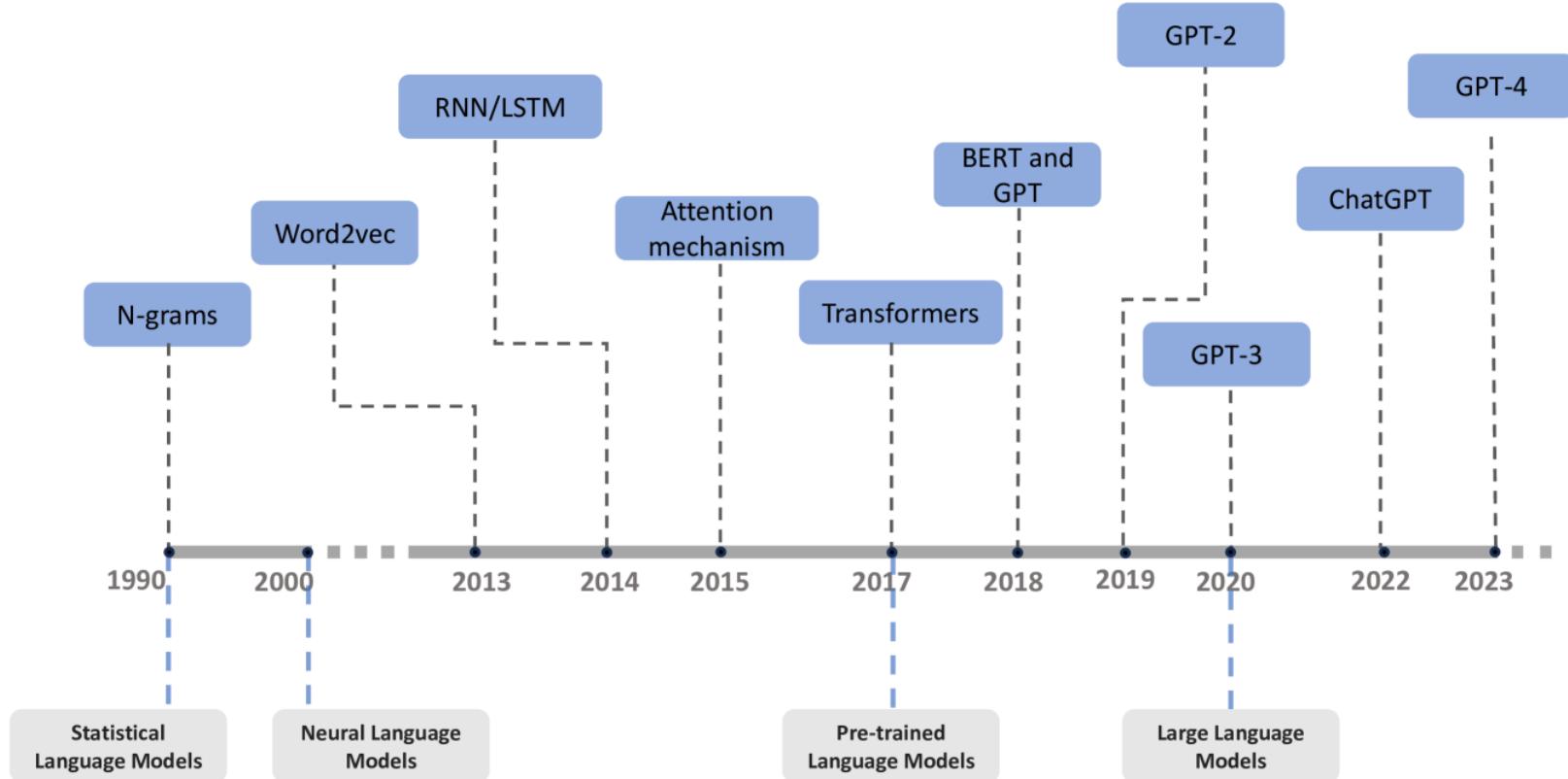
Convolution Neural Network



Demo



Language Models Journey



History, Development, and Principles of Large Language Models—An Introductory Survey, <https://arxiv.org/html/2402.06853v1>

Large Language Models

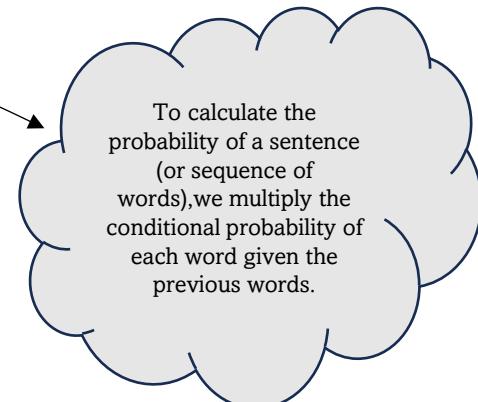
What is a Language Model

A language model is a probabilistic model that assigns a probability to a sequence of words and predicts the likelihood of the next word in a sentence, given the previous words.

Statistical Language Model (SLM):

A language model estimates the probability distribution over sequences of words. Given a sequence of words $w_1, w_2, w_3, \dots, w_n$, a language model computes:

$$P(w_1, w_2, \dots, w_n) = \prod_{i=1}^n P(w_i | w_1, \dots, w_{i-1})$$



Chapter 3: N-gram Language Models , Jurafsky, D., & Martin, J. H. (2023). *Speech and Language Processing* (3rd ed. draft). Stanford University.
<https://web.stanford.edu/~jurafsky/slp3/>

Large Language Models

Predict the next word

I want to drink a hot cup of _____

Training Corpus

1. I want to drink a hot cup of coffee
2. Every morning, I drink a hot cup of coffee before work
3. He prefers a hot cup of tea in the evening
4. She needs a hot cup of coffee to wake up
5. After dinner, they drank a hot cup of tea
6. I always start my day with a hot cup of black coffee
7. On cold days, people enjoy a hot cup of cocoa
8. I want to drink a hot cup of coffee quickly
9. They like to have a hot cup of herbal tea after yoga
10. I usually order a hot cup of coffee at Starbucks

Large Language Models

Derivation

Predict the next word

I want to drink a hot cup of _____

Training Corpus

1. I want to drink a hot cup of coffee
2. Every morning, I drink a hot cup of coffee before work
3. He prefers a hot cup of tea in the evening
4. She needs a hot cup of coffee to wake up
5. After dinner, they drank a hot cup of tea
6. I always start my day with a hot cup of black coffee
7. On cold days, people enjoy a hot cup of cocoa
8. I want to drink a hot cup of coffee quickly
9. They like to have a hot cup of milk after yoga
10. I usually order a hot cup of coffee at Starbucks

1. Expression to find the next word

$$P(w_1, \dots, w_9) \approx \prod_{i=1}^9 P(w_i | w_{i-2}, w_{i-1})$$

2. Chain rule of probability

$$P(w_1) \cdot P(w_2 | w_1) \cdot P(w_3 | w_1, w_2) \cdot P(w_4 | w_2, w_3) \cdots \cdots P(w_9 | w_7, w_8)$$

3. If we assume the next word is coffee

$$P(\text{"I want to drink a hot cup of coffee"}) = P(w_1, w_2, \dots, w_9)$$

$$= P(w_1) \cdot P(w_2 | w_1) \cdot P(w_3 | w_1, w_2) \cdots \cdots P(w_9 | w_1, \dots, w_8)$$

Phrase	Count	Probability
cup of coffee	5	= 5/10 (0.5)
cup of tea	2	= 2/10 (0.2)
cup of milk,	1	= 1/10 (0.1)
Total ("cup of X")	10	

4. The predicted word would be coffee

The Issues with Statistical Model

History, Development, and Principles of Large Language Models—An Introductory Survey

Zhibo Chu

Shenzhen Institutes of Advanced Technology, Chinese Academy
of Sciences, Shenzhen, China

University of Science and Technology of China, Hefei, China

Shiwen Ni

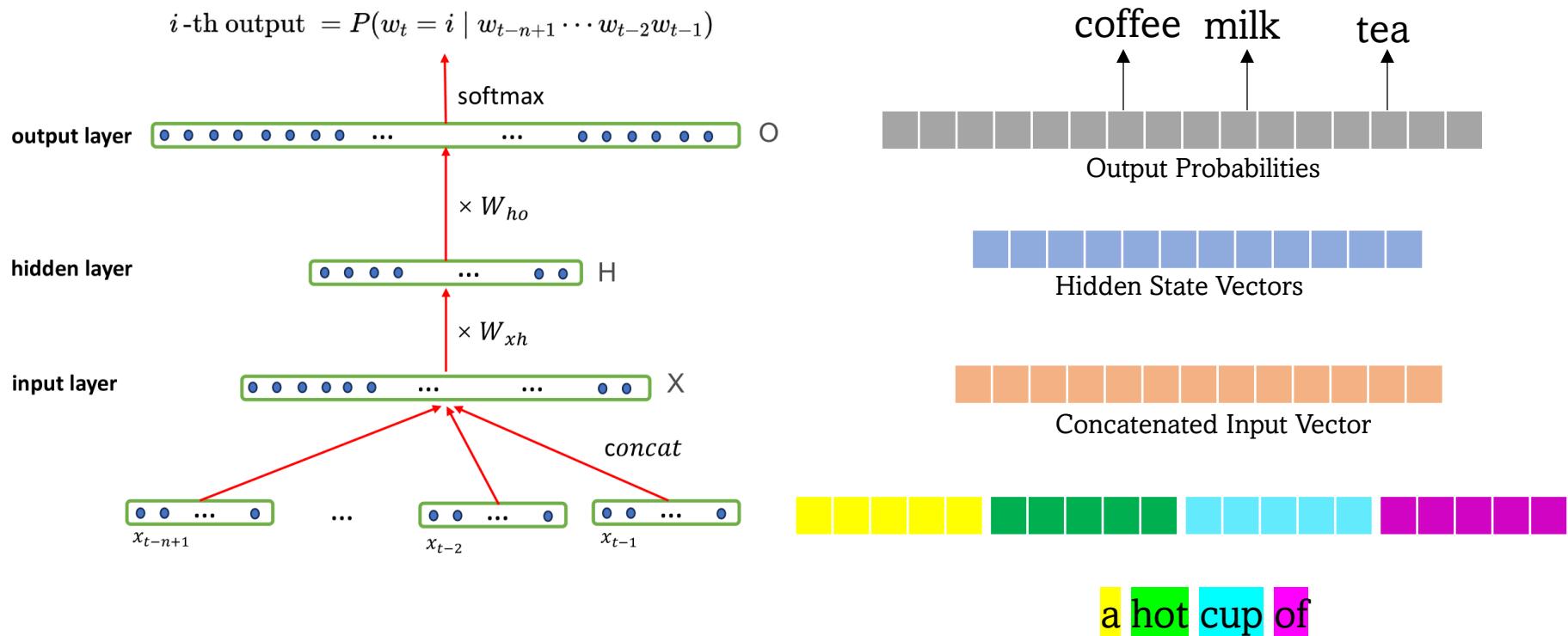
corresponding author Shenzhen Institutes of Advanced
Technology, Chinese Academy of Sciences, Shenzhen, China

conditional probabilities, it is necessary to pre-compute and save $C(X)$ required for the conditional probability computation, where X is a sentence of length n . The number of possible sentences X grows exponentially with the size of the vocabulary. For instance, with 1000 different words, there exist 1000^n potential sequences of length n . However, excessively large values of n pose storage limitations. Typically, n is confined to 2 or 3, causing each word to relate to only its first 1 or 2 preceding words, ultimately leading to a reduction in the model's accuracy.

Large Language Models

Neural Language Models

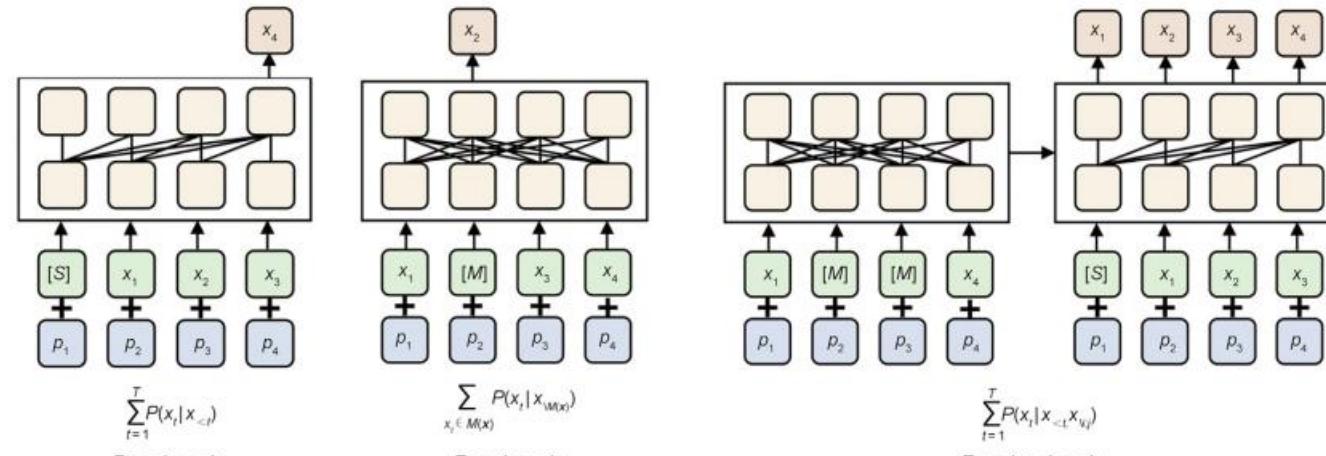
Neural Language Models: NLMs ([Bengio et al., 2000](#); [Mikolov et al., 2010](#); [Kombrink et al., 2011](#)) leverage neural networks to predict the probabilities of subsequent words within sequences. They effectively handle longer sequences and mitigate the limitations associated with small n in SLMs. Before delving into neural networks, let's grasp the concept of



Large Language Models

Large Language Model

Pre-trained Language Model: PLMs undergo initial training using an extensive volume of unlabeled text, enabling them to grasp fundamental language structures such as vocabulary, syntax, semantics, and logic — a phase termed pre-training. Subsequently, this comprehensive language model can be applied to various NLP tasks like machine translation, text summarization, and question-answering systems. To optimize its performance, models need to be trained a second time on a smaller dataset customized for a specific downstream task — a phase known as fine-tuning. This is the “pre-training and fine-tuning” learning paradigm. We can use a visual example to understand the “pre-training and fine-tuning”, as follows:



[Demo](#)

GPT

BERT

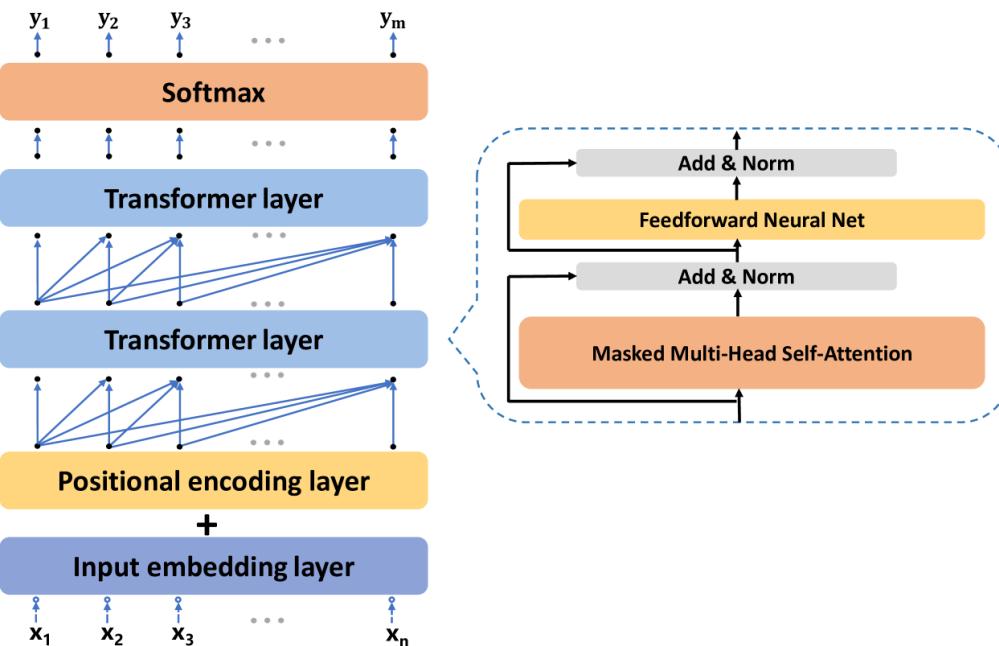
Bidirectional and Auto-Regressive
Transformers (BART)

Wang, H., Li, J., Wu, H., Hovy, E., & Sun, Y. (2023). Pre-trained language models and their applications. *Engineering*, 25, 51-65.

Large Language Models

“A Large Language Model is a transformer-based neural network trained to model the probability distribution over sequences of words or tokens, enabling tasks such as text generation, summarization, translation, and question answering.”

- Bommasani et al., 2021, *On the Opportunities and Risks of Foundation Models*



Examples: GPT-4, BERT
LLaMA, Claude, Gemini, Mistral

[Demo](#)

Large Language Models

LLM Datasets

Dataset Name	Size	Dataset Information	Languages	URL
Common Crawl	Petabyte-scale	Web pages, blogs, news articles, forums	100+	https://commoncrawl.org
The Pile	825 GB	Academic papers, books, GitHub, StackExchange, Wikipedia, PubMed, etc.	Primarily English	https://pile.eleuther.ai
Wikipedia	~20 GB (English)	Encyclopedic articles	300+	https://dumps.wikimedia.org
OpenWebText2	~40 GB	High-quality content from web links in Reddit	Primarily English	https://github.com/EleutherAI/openwebtext2
RedPajama	~1.2 TB	Common Crawl, C4, Books, GitHub, Wikipedia, StackExchange	Primarily English	https://www.together.xyz/blog/redpajama
The Stack	3.1 TB	Source code from GitHub in 30+ programming languages	30+ (programming languages)	https://huggingface.co/datasets/bigcode/the-stack
arXiv + PubMed	10+ GB	Scientific papers in physics, math, medicine, biology	Primarily English	https://pubmed.ncbi.nlm.nih.gov/download https://www.kaggle.com/datasets/Cornell-University/arxiv

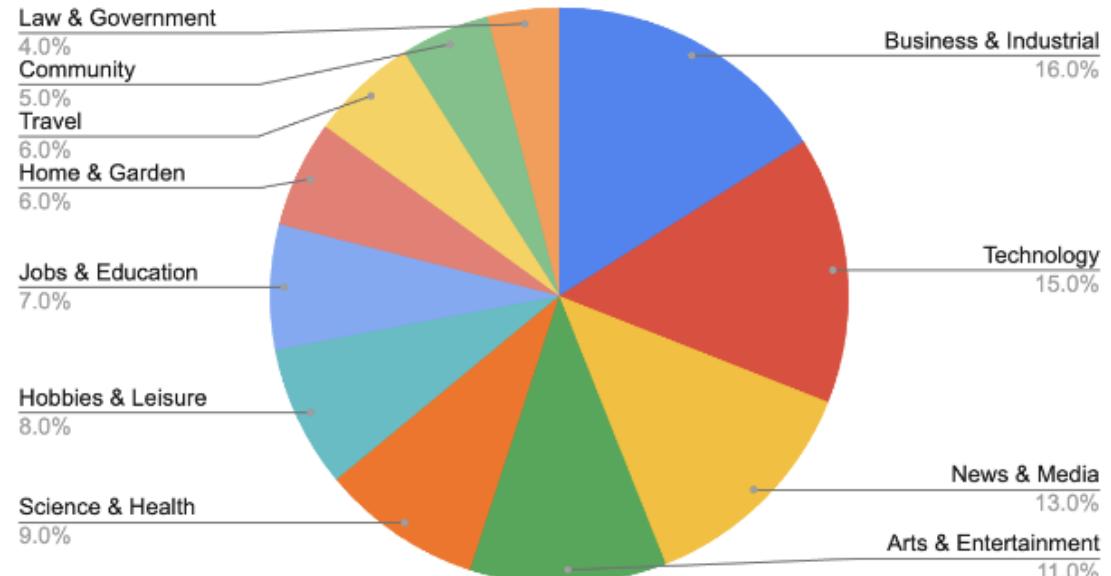
Common Crawl Dataset

To understand models, we have to start with data. Foundation models need a lot of data. Knowing what data a model is trained on gives important clues about what it can do.

For example, a common source for training data is Common Crawl, created by a nonprofit organization that sporadically crawls websites on the Internet.

Language	Speakers (million)	% world population	% in Common Crawl	Ratio: World population / Common Crawl
Punjabi	113	1.41%	0.0061%	231.56
Swahili	71	0.89%	0.0077%	115.26
Urdu	231	2.89%	0.0274%	105.38
Kannada	64	0.80%	0.0122%	65.57
Telugu	95	1.19%	0.0183%	64.89
Gujarati	62	0.78%	0.0126%	61.51
Marathi	99	1.24%	0.0213%	58.10
Bengali	272	3.40%	0.0930%	36.56
English	1452	18.15%	45.88%	0.40

Distribution of domains in the C4 dataset



Large Language Models Evaluation

Total Cats = 6 + 1

Total Dogs = 2 + 12

True Positive & True Negative: When Actual and Predicted values are same.

TP = 6, TN = 11

False Positive: When the actual Value was negative “dog” and the system predicted positive “cat”.

FP = 2

False Negative: When the actual value was “cat” and the system is predicted it “dog”

FN = 1

		PREDICTED VALUES	
		Positive (CAT)	Negative (DOG)
ACTUAL VALUES	Positive (CAT)	TRUE POSITIVE 6 	FALSE NEGATIVE 1 
	Negative (DOG)	FALSE POSITIVE 2  TYPE I ERROR	TRUE NEGATIVE 11  YOU ARE NOT A CAT

Confusion Matrix

Large Language Models Evaluation

TP = 6, TN = 11

FP = 2, FN = 1

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall}$$

		PREDICTED VALUES	
		Positive (CAT)	Negative (DOG)
ACTUAL VALUES	Positive (CAT)	TRUE POSITIVE 6	FALSE NEGATIVE 1
	Negative (DOG)	FALSE POSITIVE 2	TRUE NEGATIVE 11
		YOU ARE A CAT	YOU ARE A DOG
		TYPE II ERROR	TYPE I ERROR
		YOU ARE NOT A CAT	YOU ARE A CAT

Confusion Matrix

Large Language Models Evaluation

BLEU (Bilingual Evaluation Understudy) – compares n -gram overlaps between prediction and reference

Papineni, K., Roukos, S., Ward, T., & Zhu, W. J. (2002).

Let's calculate it for bigram

$$\text{BLEU} = \text{BP} \cdot \exp \left(\sum_{n=1}^N w_n \log p_n \right).$$

$$\text{BLEU-2} = \text{BP} \cdot \exp \left(\frac{1}{2} (\log p_1 + \log p_2) \right)$$

Actual: The cat is on the mat

Unigram: the, cat, is, on, the, mat

Bigram: the cat, cat is, is on, on the, the mat

Predicted: The cat sat on the mat

Unigram: the, cat, sat, on, the, mat

Bigram: the cat, cat sat, sat on, on the, the mat $p2 = \frac{3}{6}$

$$p1 = \frac{5}{6}$$

$$\text{BLEU} - 2 = 1 * e^{(\frac{1}{2}(\log \frac{5}{6} + \log \frac{3}{6}))}$$

$$0.645$$

BP stands for **Brevity Penalty**. It is used to penalize machine-generated text that is **too short** compared to the reference

BLEU score is precision-oriented (counts how many n-grams match), but without a length penalty, a model could **cheat** by just outputting short sequences.

BP solves this by lowering the BLEU score when the generated output is shorter than the reference.

$$\text{BP} = \begin{cases} 1 & \text{if } c > r \\ e^{(1-\frac{r}{c})} & \text{if } c \leq r \end{cases}$$

Large Language Models Evaluation

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) Score

Lin, C.-Y. (2004).

ROUGE: A Package for Automatic Evaluation of Summaries

ROUGE (Recall-Oriented) is used in summarization. The most common are:

- ROUGE-1: Overlap of unigrams

$$\text{ROUGE-1(Precision)} = 5/6, \text{ROUGE-1(Recall)} = 5/6, \text{ROUGE-1(F1)} = \frac{2 * \frac{5}{6} * \frac{5}{6}}{\frac{5}{6} + \frac{5}{6}}$$

- ROUGE-2: Overlap of bigrams

$$\text{ROUGE-2(Precision)} = 3/6, \text{ROUGE-1(Recall)} = 3/6, \text{ROUGE-1(F1)} = \frac{2 * \frac{3}{6} * \frac{3}{6}}{\frac{3}{6} + \frac{3}{6}}$$

- ROUGE-L: Longest Common Subsequence (LCS)

Longest Sequence (5) = The cat [mismatch/gap] on the mat

$$\text{ROUGE-L(Precision)} = 5/6, \text{ROUGE-1(Recall)} = 5/6, \text{ROUGE-1(F1)} = \frac{2 * \frac{5}{6} * \frac{5}{6}}{\frac{5}{6} + \frac{5}{6}}$$

Some Benchmarks

Model	MMLU	HumanEval	GSM8K	TruthfulQA
GPT-4	86.40%	88.00%	94.00%	59.00%
Claude 2	81.60%	71.00%	88.00%	58.00%
Gemini 1.5 Pro	84.00%	83.00%	92.00%	62.00%
Claude 3 Opus	88.70%	90.00%	95.00%	68.00%
GPT-3.5	70.00%	48.10%	57.10%	47.00%
LLaMA 2 70B	79.00%	67.00%	83.00%	52.00%
Mixtral 8x7B	84.10%	74.00%	87.00%	58.50%
Mistral 7B	70.00%	55.00%	65.00%	47.00%
Command R+	75.20%	60.50%	78.00%	53.10%
Gemma 7B	65.00%	45.00%	58.00%	41.00%

MMLU (Massive Multitask Language Understanding)

- **What it tests:** Knowledge and reasoning across 57 academic subjects like history, law, math, medicine, etc.
- **Use case:** Checks how well a model performs on real-world, high school to graduate-level exams.

HumanEval

- **What it tests:** Code generation and reasoning.
- **Use case:** Given a prompt (like a function definition), the model needs to generate correct Python code that passes test cases.

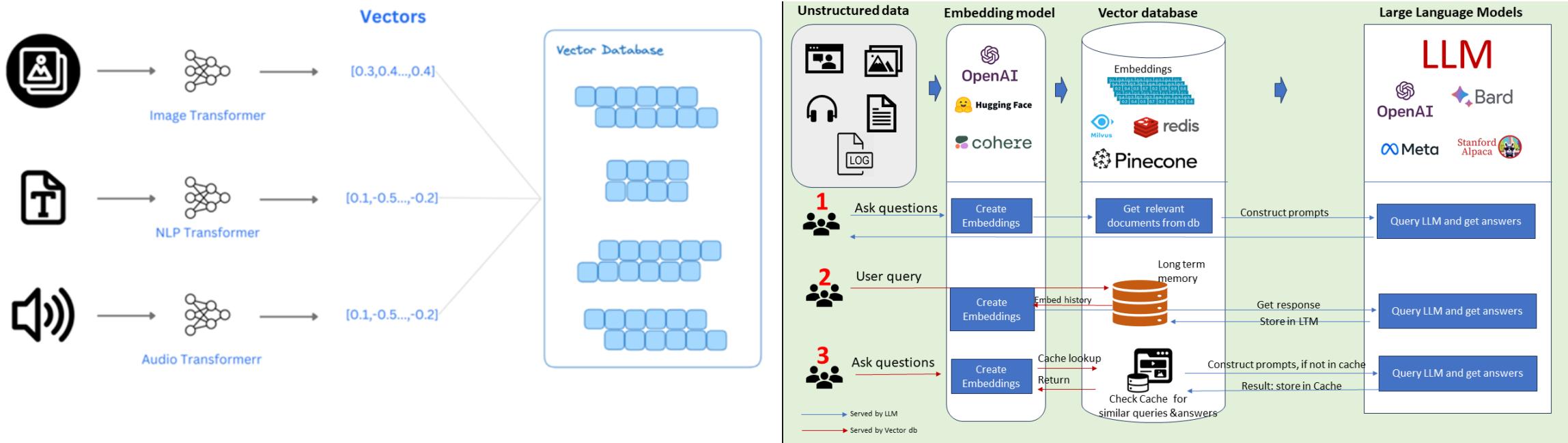
GSM8K (Grade School Math 8K)

- **What it tests:** Basic arithmetic and word problem-solving.
- **Use case:** Models solve grade-school level math problems using step-by-step reasoning.

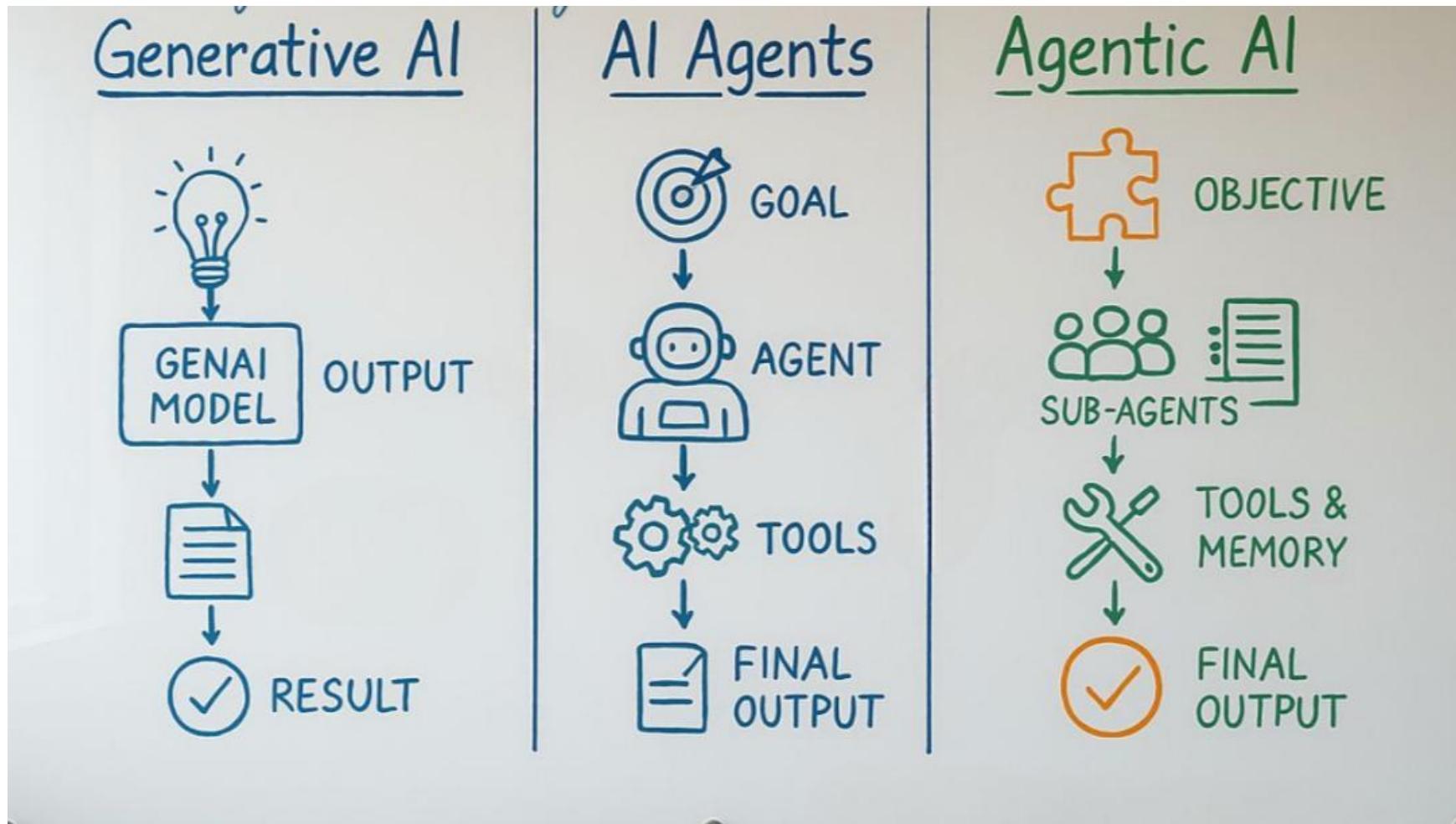
TruthfulQA

- **What it tests:** The ability to give **truthful** answers, especially in tricky or misleading questions.
- **Use case:** The model is asked questions where giving a common but false answer is easy (e.g., urban myths).

Retrieval Augmented Generation



The Differentiation



Resources: Hugging Face

The image displays a grid of eight cards, each representing a different AI model or demo available on the Hugging Face platform. The cards are arranged in two rows of four. Each card includes the model name, a brief description, the host organization, and the date it was last updated.

Model / Demo	Description	Host Organization	Last Updated
Qwen Image Edit 2509	Generate edited images based on user prompts	Qwen	3 days ago
Wan2.2 Animate	Wan2.2 Animate	Wan-AI	about 2 hours ago
Photo Mate i2i	Image manipulation with Kontext adapters	prithivMLmods	5 days ago
Qwen3 TTS Demo	Generate speech from text with voice options	Qwen	6 days ago
granite-docling-258M demo	Convert images to structured documents and answer questions	ibm-granite	10 days ago
Qwen3 Omni Demo	Interact with a multimodal chatbot using text, audio, images, or video	Qwen	6 days ago
VoxCPM Demo	VoxCPM	openbmb	10 days ago
Ostris' AI Toolkit	Train FLUX, Qwen and Wan LoRAs with Ostris Ai Toolkit	multimodalart	8 days ago

Ollama



Cloud models are now available in Ollama

**Chat & build with
open models**

[Download](#)