Data Science, AI, and Machine Learning in Public Health using R

LLM Overview

December 2025

Presented By: Wronski Associates

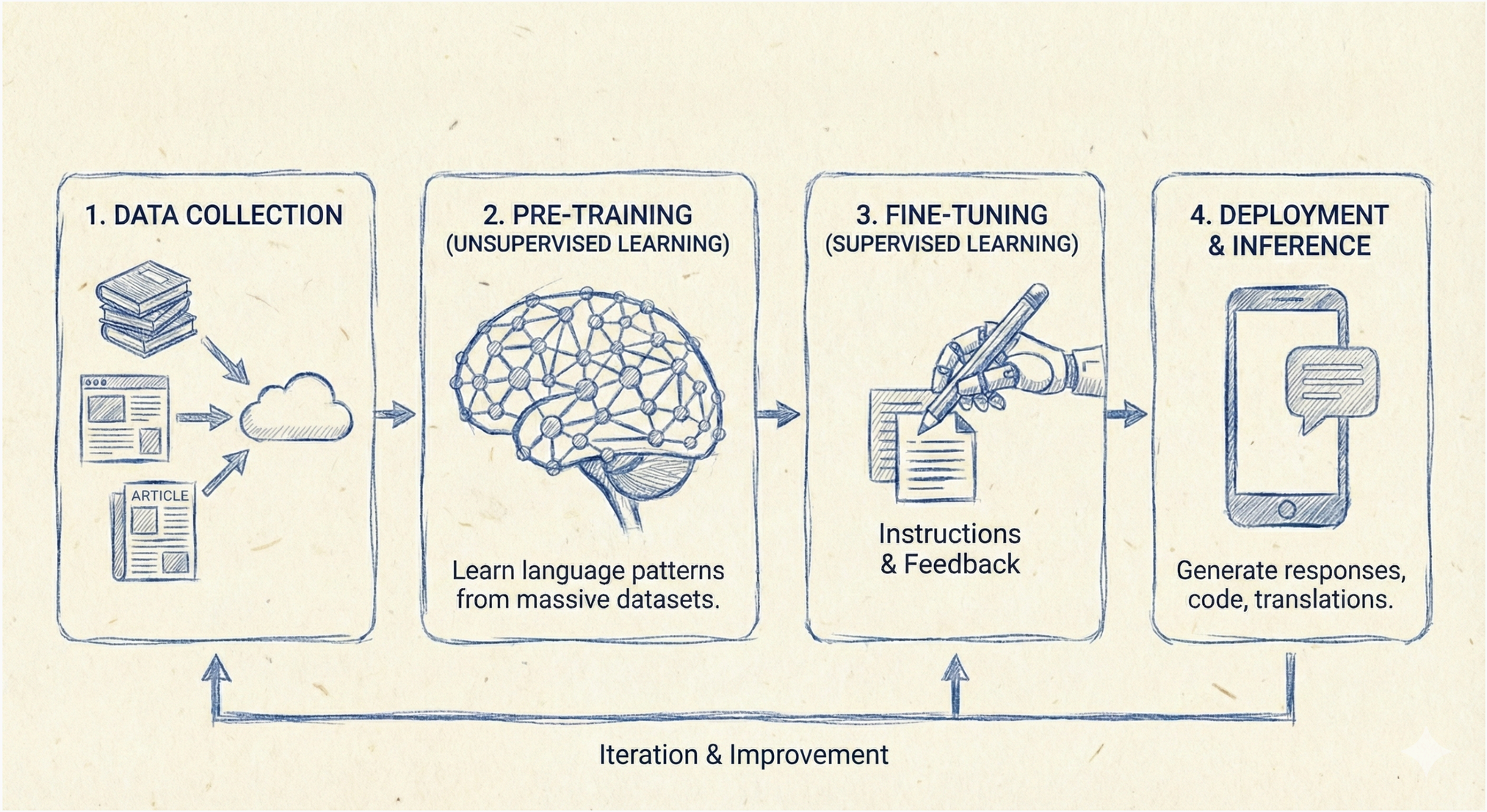
Large Language Models (LLMs)

A very large text-prediction system that has learned patterns from a huge corpus of human writing. An LLM:

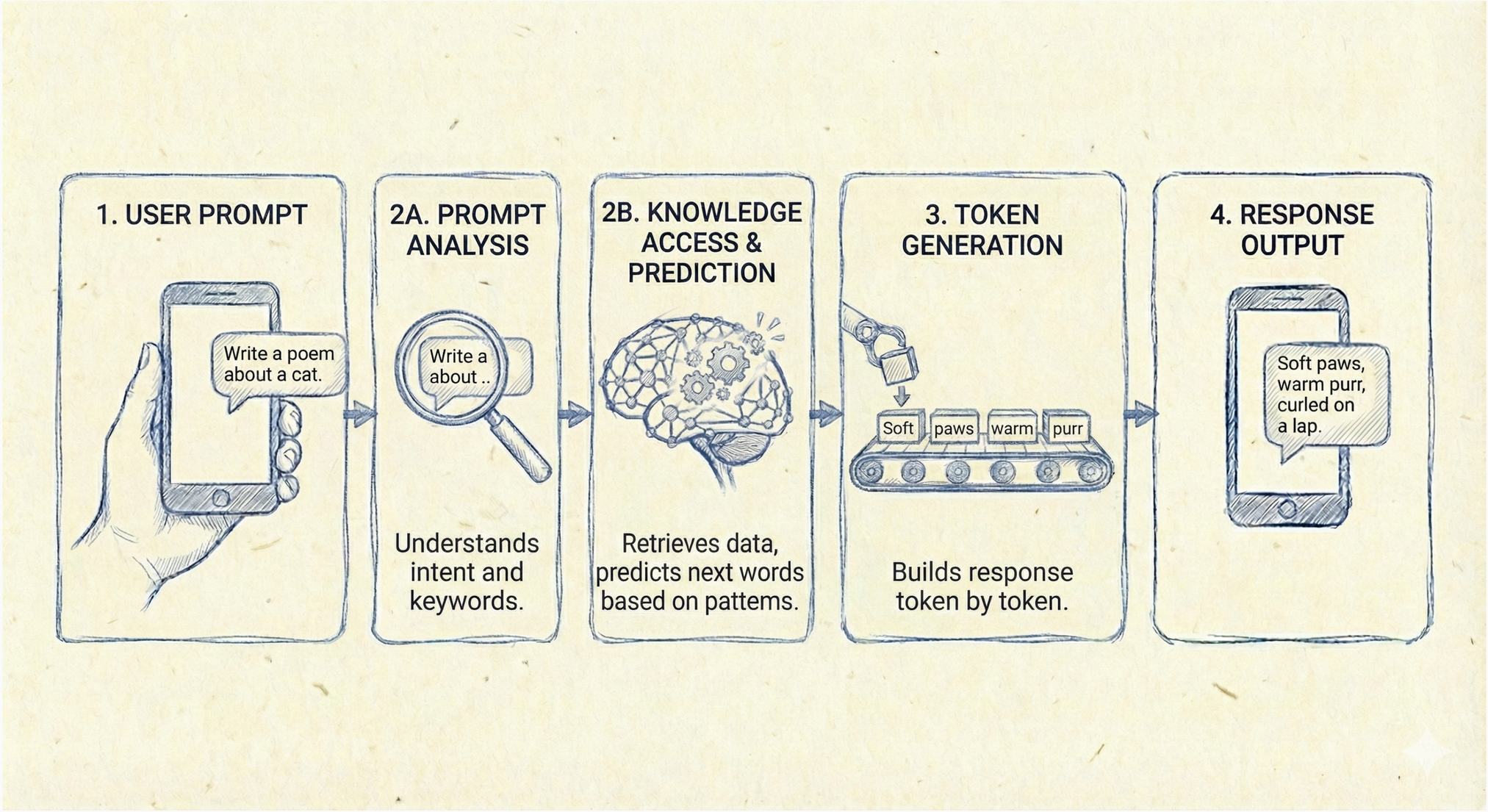
What it actually does:

- Has learned from text
  - The LLM is trained on vast collections of text: news, books, websites, social media, technical docs, etc.
  - During training, it is repeatedly asked:
    - “Given this text so far, what word (or token) is likely to come next?”
  - Over time, it becomes very good at predicting these next words.
- Uses patterns rather than rules
  - It does not have hand-written rules like,
    - “If a user asks about Congress, mention three branches of government.”
  - Instead, it has learned statistical patterns:
    - how people typically talk about government,
    - common associations, arguments, and rhetorical moves,
    - which words tend to appear together in which contexts.
- Generates a response as imitation
  - When you ask a question, the LLM generates a response that imitates the kinds of answers it has seen in its training data.

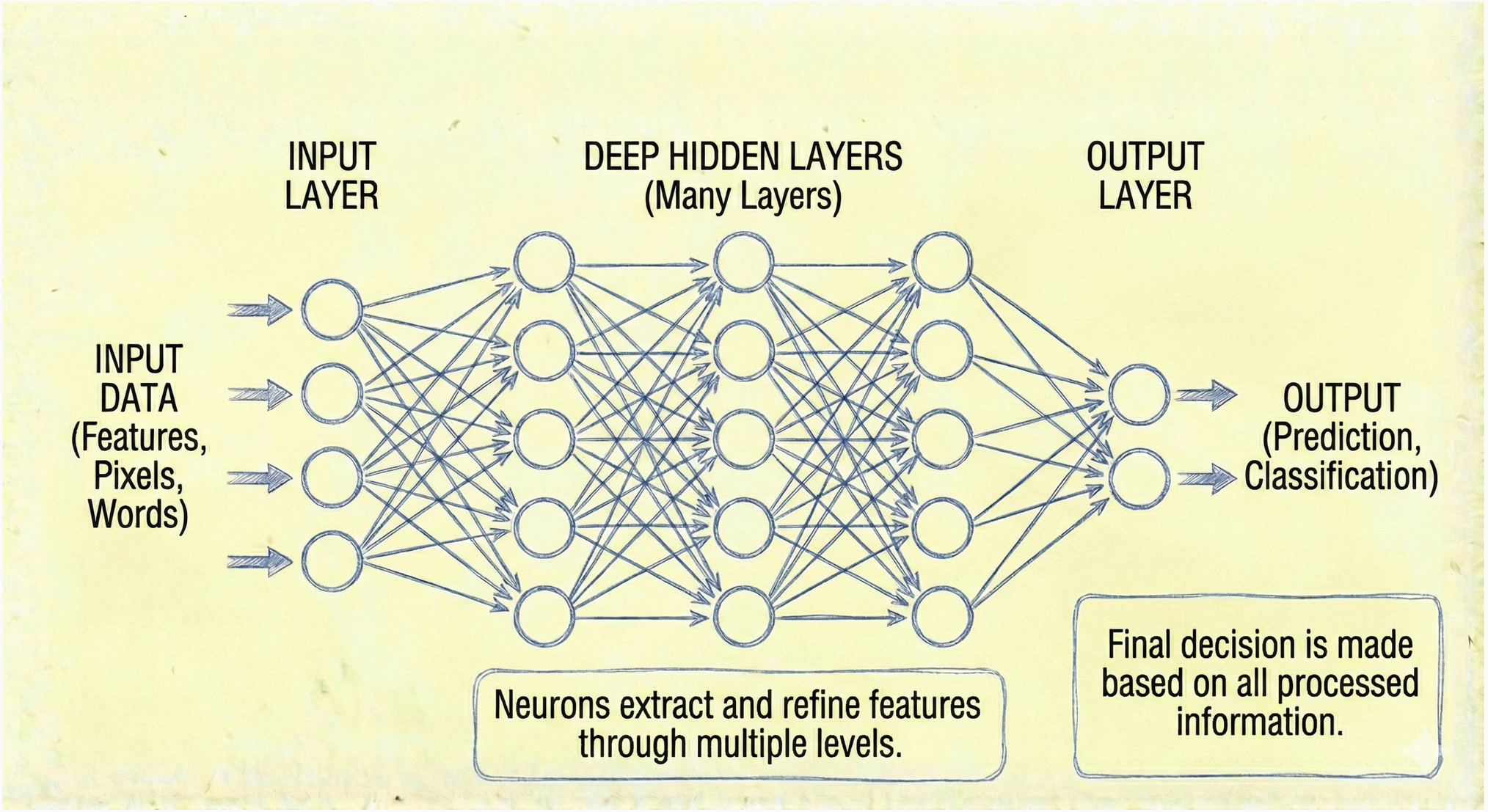
Building the model



Using the model



Deep Learning



Note: A neural network is a specific kind of machine learning model inspired (loosely) by the brain. Deep learning is a subfield of machine learning

Neural Network models

Model type	One-sentence description
Convolutional Neural Network (CNN)	A deep learning model that uses sliding filters to detect visual patterns, making it especially good for image data.
Recurrent Neural Network (RNN) / LSTM	A deep learning model designed for sequence data that processes inputs step by step while keeping a short memory of what came before.
Transformer	A deep learning model that uses attention to look at all parts of a sequence at once, making it powerful for language and other complex sequential data.

Deep Learning/Neural Networks/Transformers

Concept	What it is	How it relates to the others
Neural networks	A family of models built from connected layers of simple units (“neurons”) whose weights are learned from data.	Deep learning models and Transformers all live inside this broader family; they are specific kinds of neural networks.
Deep learning	A way of using large, multi-layer neural networks with lots of data and compute to learn rich internal features automatically.	“Deep learning” is the practice or subfield that typically uses deep neural networks (including CNNs, RNNs, Transformers).
Transformers	A modern neural network architecture that uses attention to let each token look at all others in a sequence at once.	Transformers are one specific type of deep neural network and are the backbone of most current large language models.

Transformers

What Is a Transformer Model?

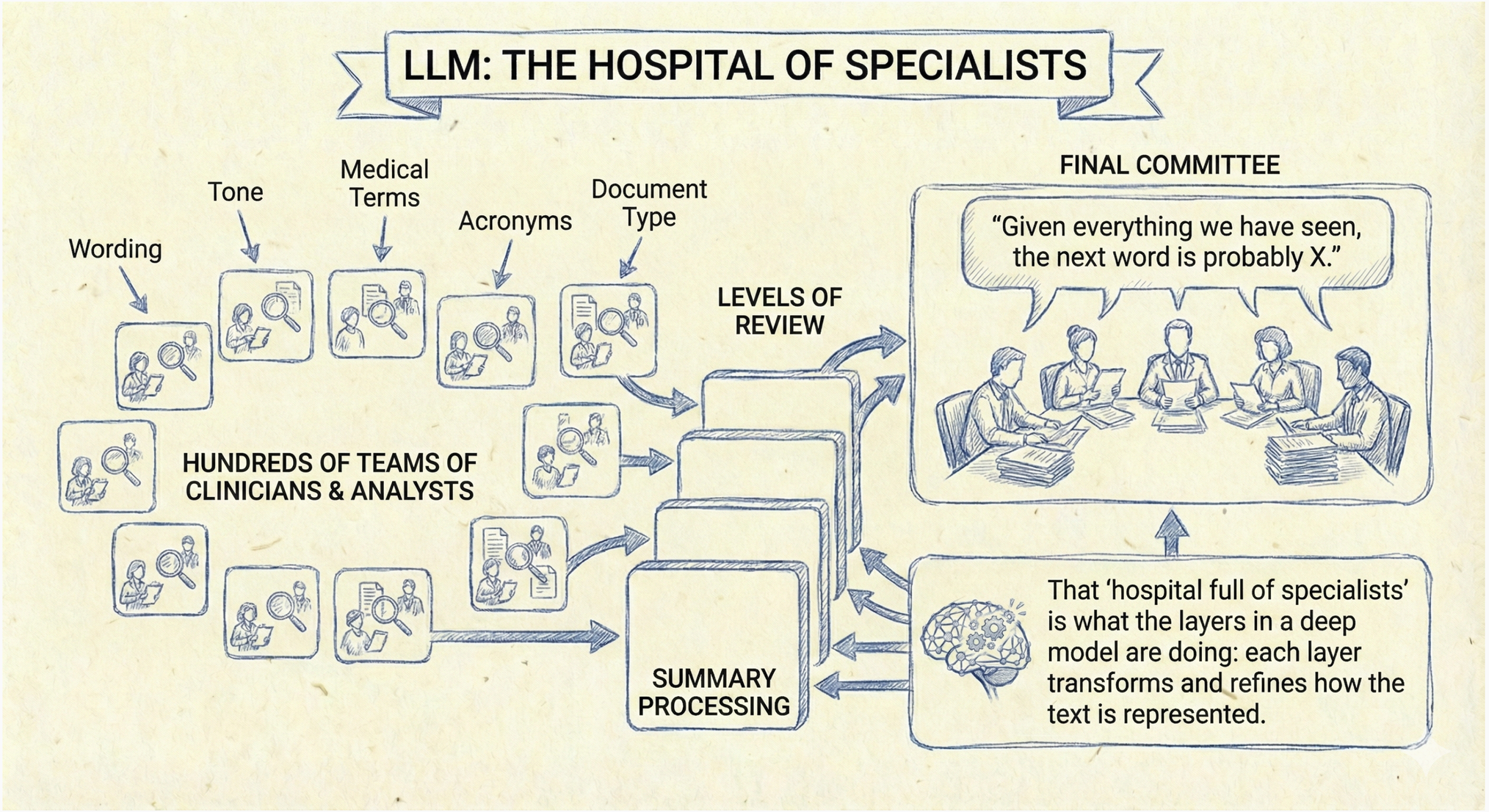
Plain English definition: A Transformer is a modern type of neural network that reads text (or other sequences) and figures out which parts matter most to each decision using a mechanism called "attention."

What it can do: Transformers power tools like ChatGPT that can summarize long reports, draft responses, classify free-text fields, and answer questions over guidelines or surveillance documentation.

How to picture it Instead of reading only left-to-right, a Transformer can look across the entire text at once, noticing patterns like "this phrase in the abstract is connected to that table in the results."

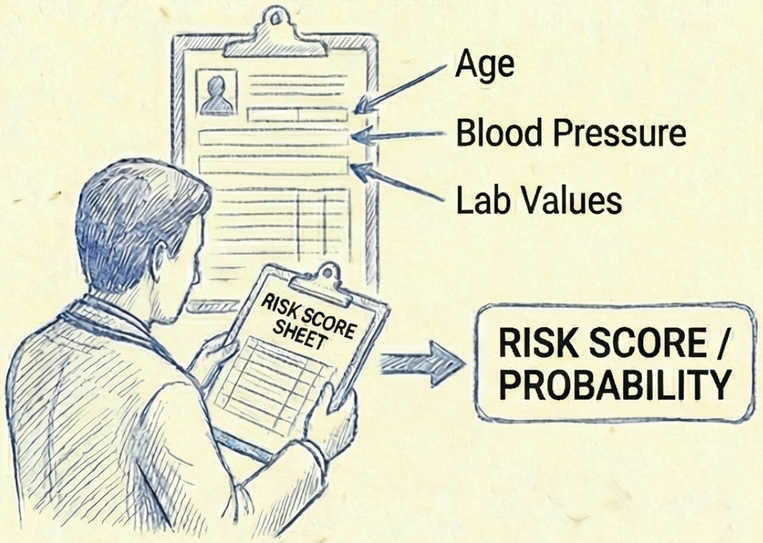
Key limitations to remember: They can sound confident but be wrong, they learn from whatever text they were trained on (including bias and gaps), and they need strong guardrails, human review, and clear use cases.

Neural network layers



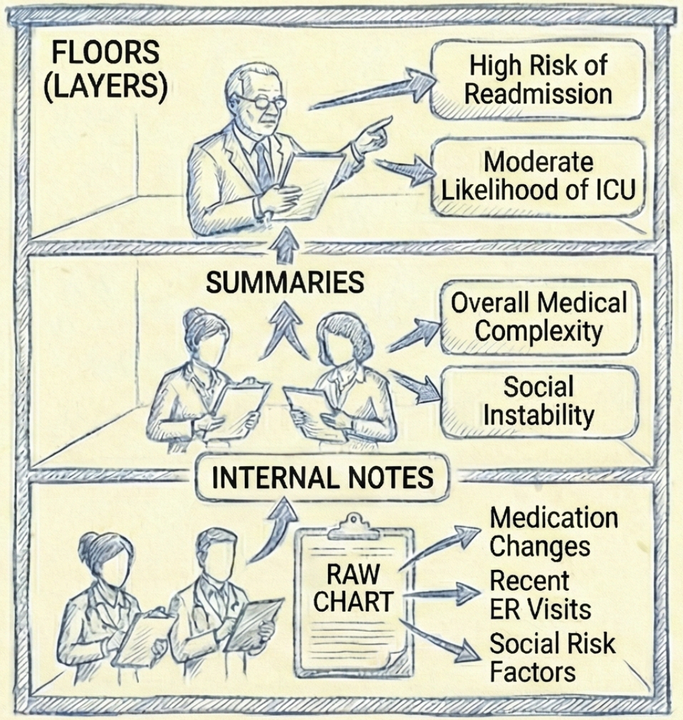
A second view

LINEAR/LOGISTIC REGRESSION



One experienced clinician, one score, one answer.

NEURAL NETWORK: THE CARE TEAM & HOSPITAL FLOORS



LAYER 3
(HIGHER FLOOR CLINICIANS & FINAL DECISION)
Process summaries for abstract decisions and final answers.

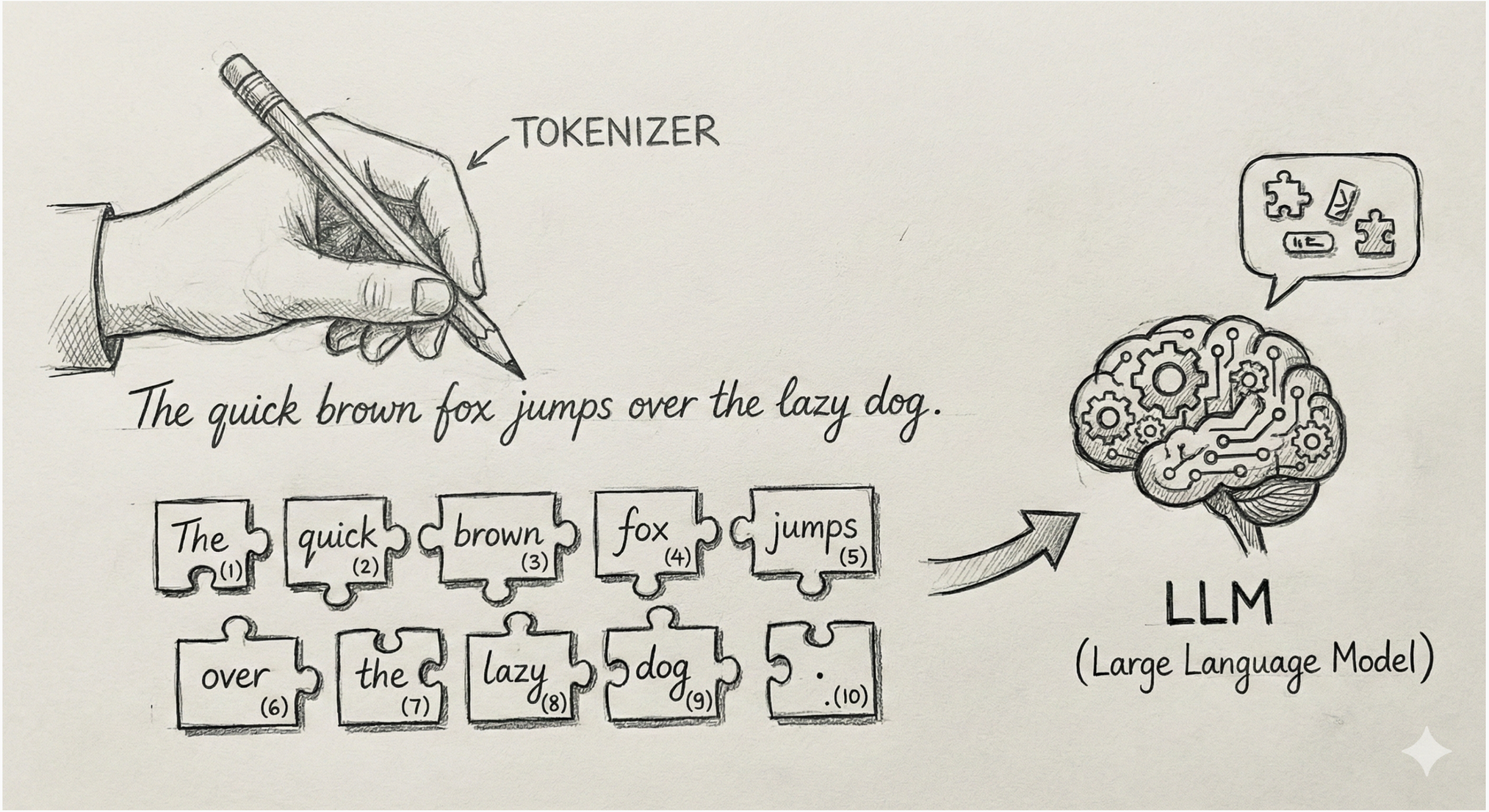
LAYER 2
(SECOND ROW CLINICIANS)
Read notes, summarize complexity and social factors.

LAYER 1
(FIRST ROW CLINICIANS)
Each focuses on something different, creating internal notes.

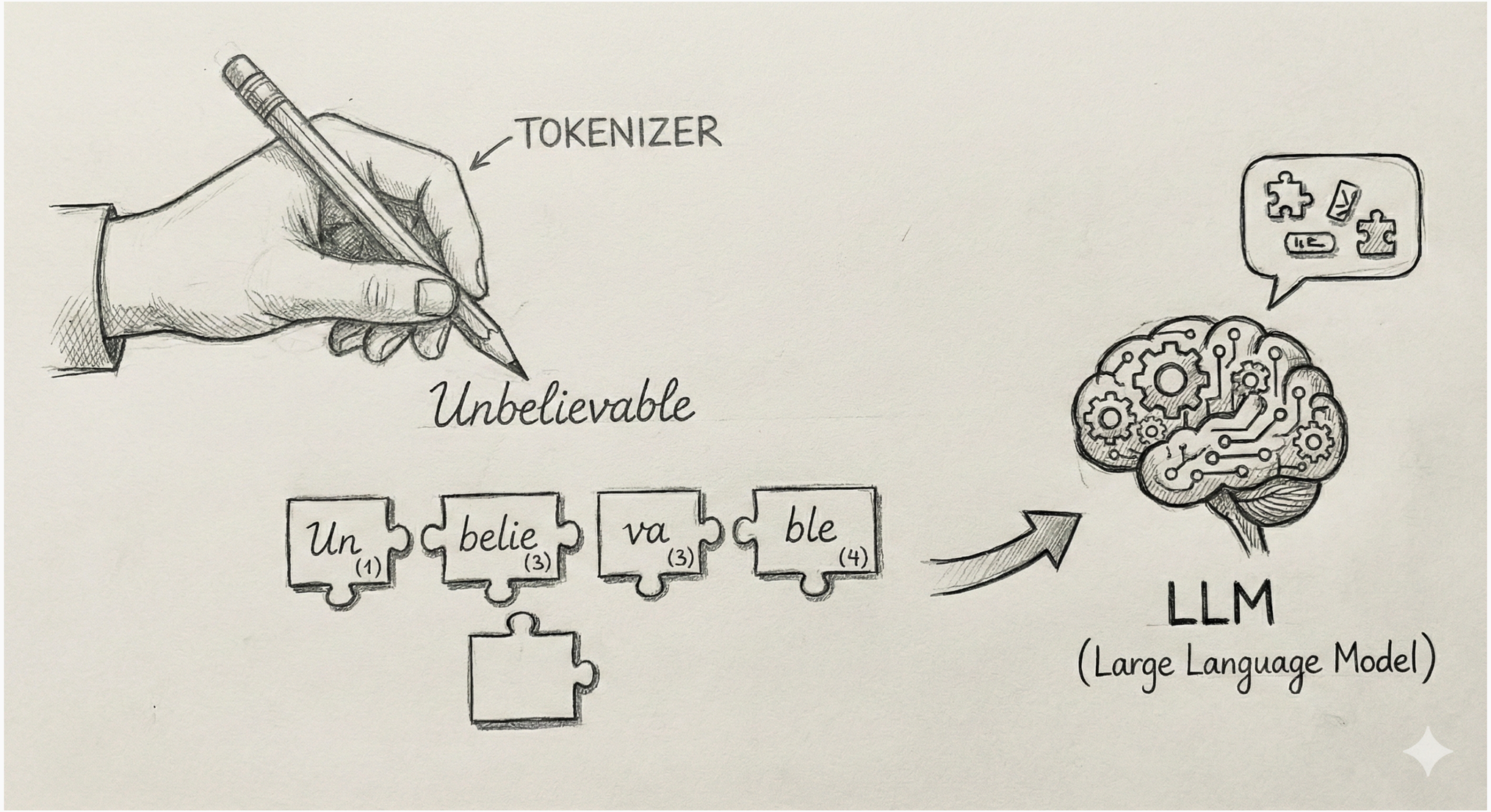
Neural Network = Many clinicians arranged by floors. Data flows up: chart → notes → summaries → final decision.

Tokenization

- Incoming text is split into small units called **tokens** (pieces of words, punctuation, etc.).
- The model does not see words directly, only these tokens.

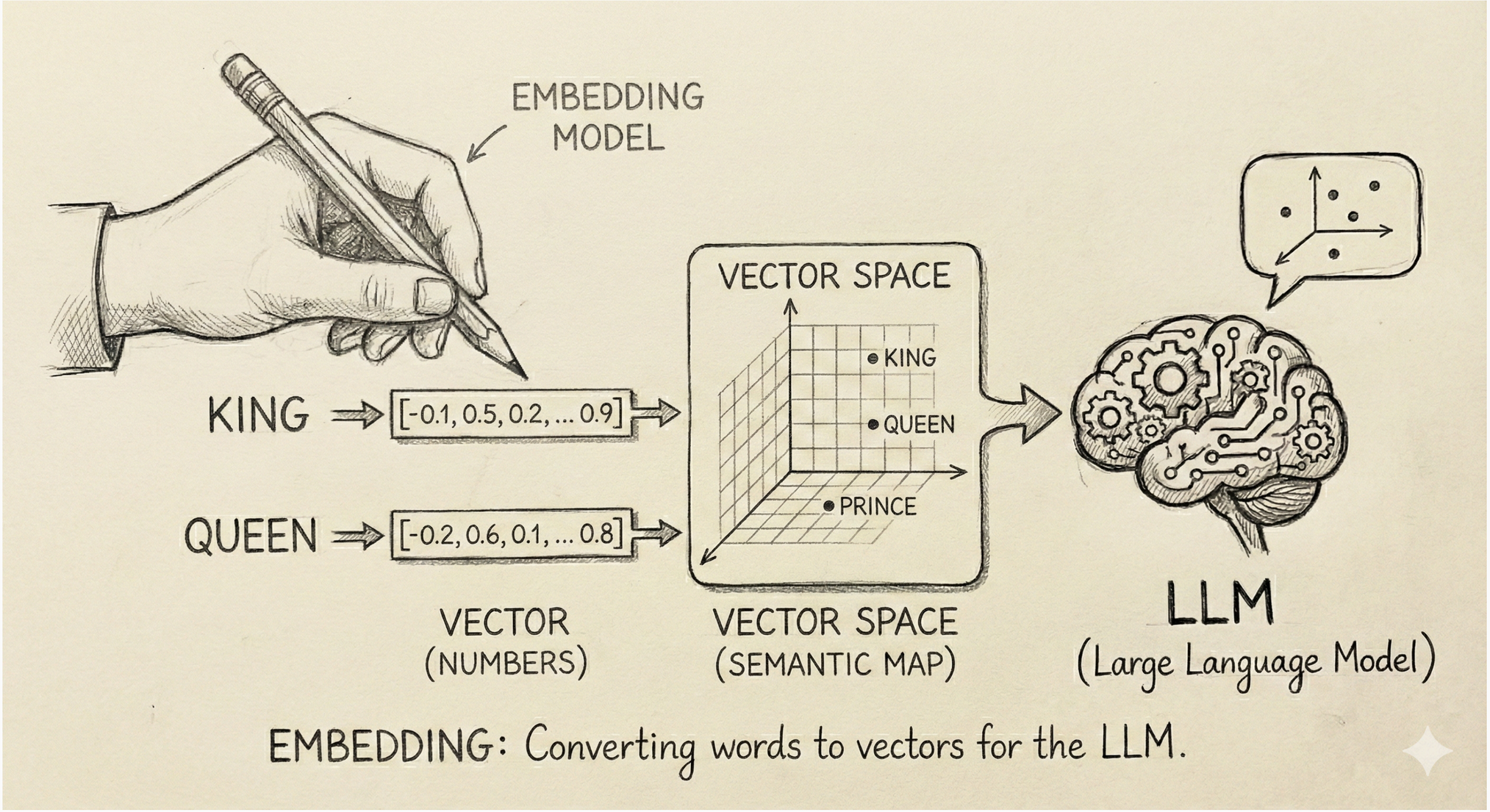


Tokenization



Embedding

- Each token is converted into a vector (an **embedding**).



Inside the model, every token is turned into one or more vectors. The token itself is just a symbolic ID; the model represents it as a vector.

Embedding

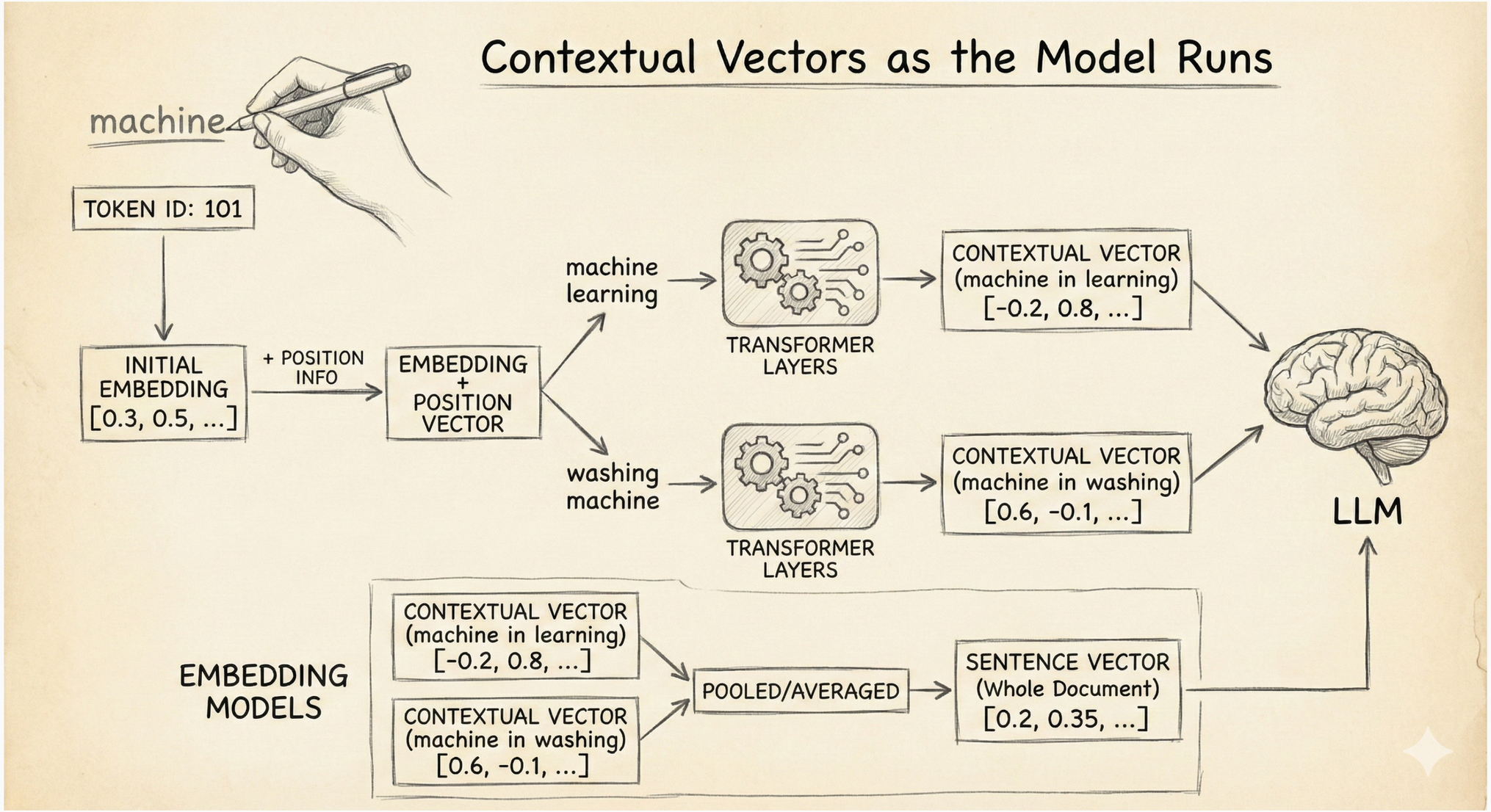
For embeddings / LLM vectors, usually we do not know what each dimension “means” in plain language:

- The model learns a big set of latent features that humans did not name in advance.
- Each value is like the amount of that latent ingredient.
- Any single number by itself is hard to interpret; what matters is:
  - the pattern across all values, and
  - how that vector relates to other vectors (similarity / distance).

Note

A vector is a list of numbers; each number is a coordinate along some feature axis. In embeddings, these values are **learned features**, so we mostly care about the whole vector’s position relative to others, not the meaning of each individual value.”

Contextual Vectors



Once initial embeddings are created:

- Models add position information (so the model knows order).

Inference (What You Use Day-to-Day)

- The trained weights are fixed.
- You send a prompt; the model runs forward passes only (no learning at that moment).
- Main concerns:
  - Latency.
  - Throughput.
  - Cost per token.

Not a Database or Rules Engine

An LLM is **not**:

- A database:
  - No explicit tables.
  - No reliable, queryable "stored facts" in the usual sense.
- A rules engine:
  - No hand-coded "if condition X then do Y" rules for every behavior.

Instead:

- Knowledge is distributed in the model weights as statistical patterns.
- It produces outputs that **look** like plausible continuations of the input text.

A Probabilistic Text Engine

- It generates text that is statistically likely given the prompt and its training.
- It can:
  - Write code.
  - Explain concepts.
  - Summarize documents.
  - Draft emails or reports.
- It can also produce fluent but incorrect answers, because it optimizes for "sounding right," not "being true."