PHP
fwdays

CHAT WITH YOUR PRIVATE DATA USING LLAMA3 AND LLPHANT IN PHP

Enrico Zimuel

Tech Lead & Principal Software Engineer at Elastic

elastic

Agenda

- Generative AI
- Deep neural network
- Large Language Model
- Prompt engineering
- Retrieval Augmented Generation (RAG)
- Embedding and Vector Search
- Llama3.1 and LLPhant for PHP

Image generated using dall-e-3

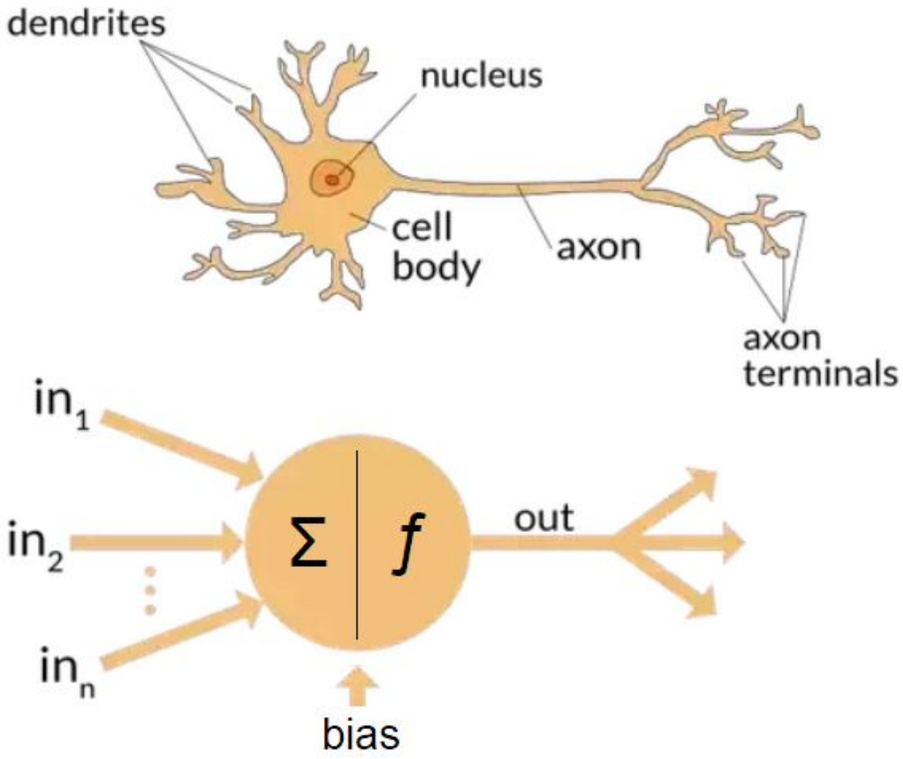
elastic

Generative AI

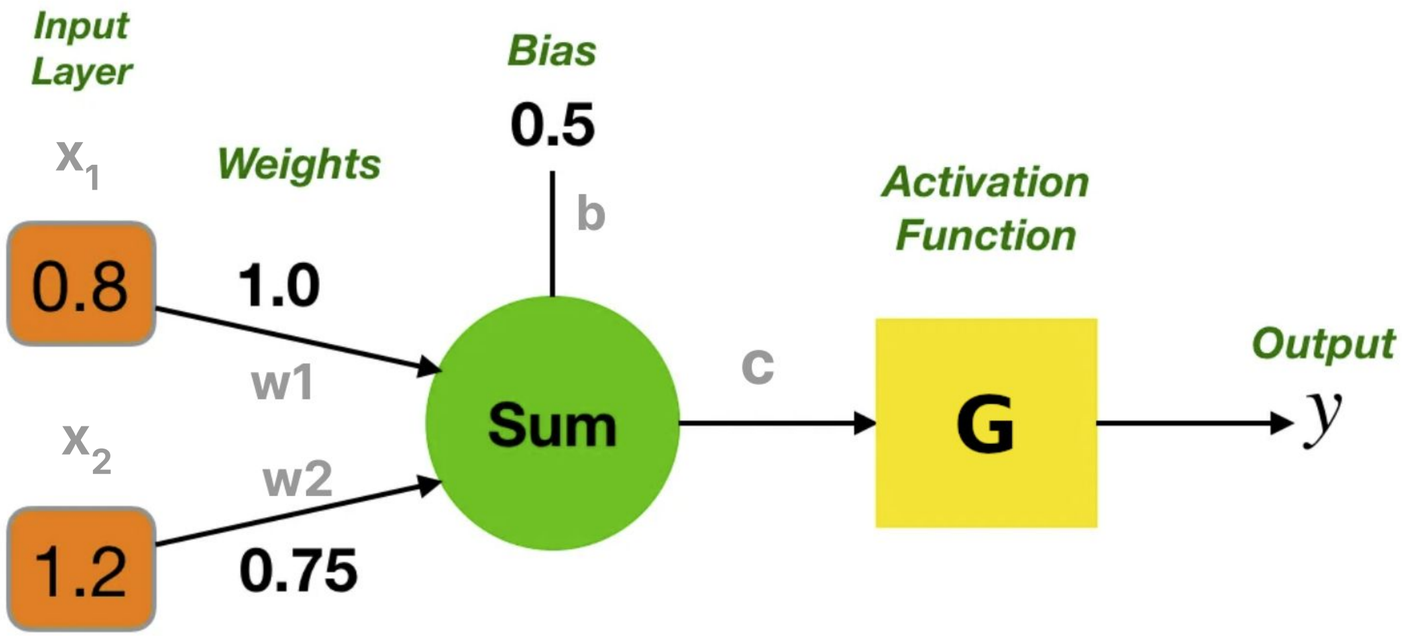
- **Generative Artificial Intelligence** (GenAI) is a subset of **deep learning** capable of generating text, images, or other media, using generative models
- GenAI models **learn the patterns and structure** of their input training data and then generate new data that has **similar characteristics**
- It’s used in many industries: art, writing, coding, healthcare, finance, gaming, marketing, etc
- The global generative ai market was valued at $10.5 billion in 2022, and is projected to reach $191.8 billion by 2032

Neural Network

- A **neural network** is a method that teaches computers to **process data** in a way that is inspired by the human brain
- Collection of **nodes** (artificial neurons) with inputs and outputs. A node computes some non-linear function of the sum of its inputs
- The nodes are collected in **layers**
- If the number of layers > 3 we call it **deep learning network**

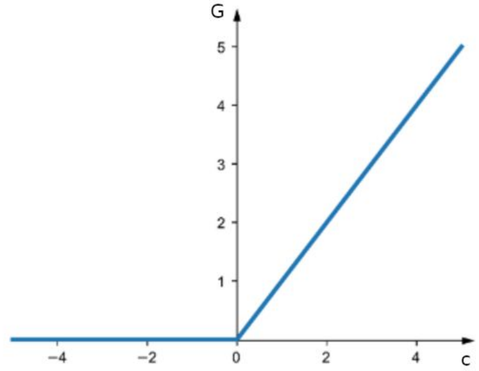


Single layer neuron



Example: ReLU activation

$G(c) = c \geq 0 \ ? \ c : 0$



$c = x_1 * w_1 + x_2 * w_2 + b$

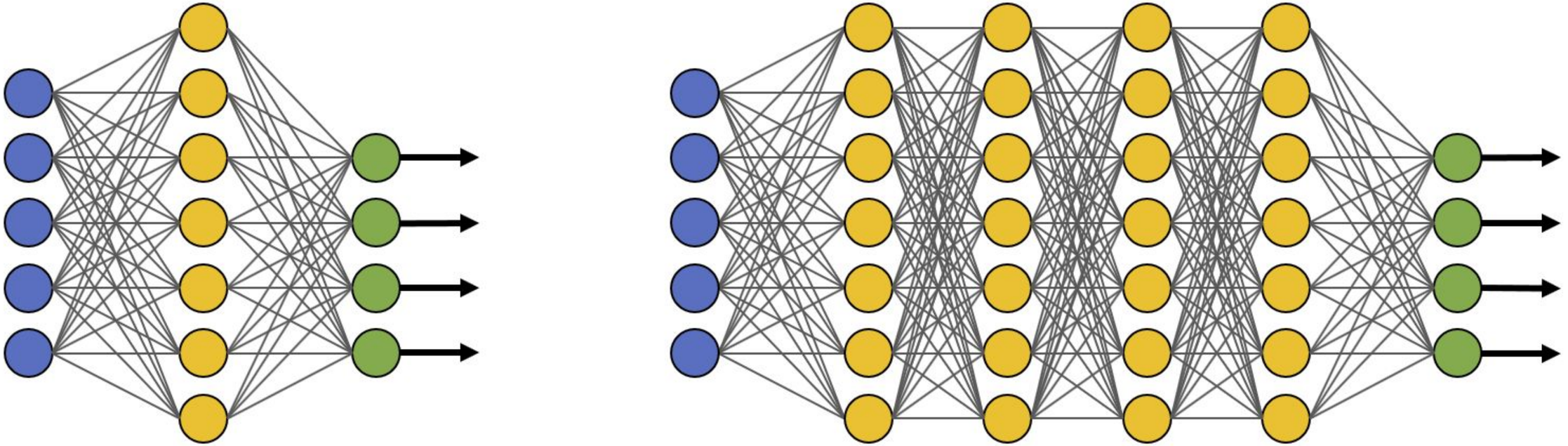
$= 0.8 * 1.0 + 1.2 * 0.75 + 0.5$

$= 2.2$

$y = G(c) = G(2.2) = 2.2$

Simple Neural Network

Deep Learning Neural Network

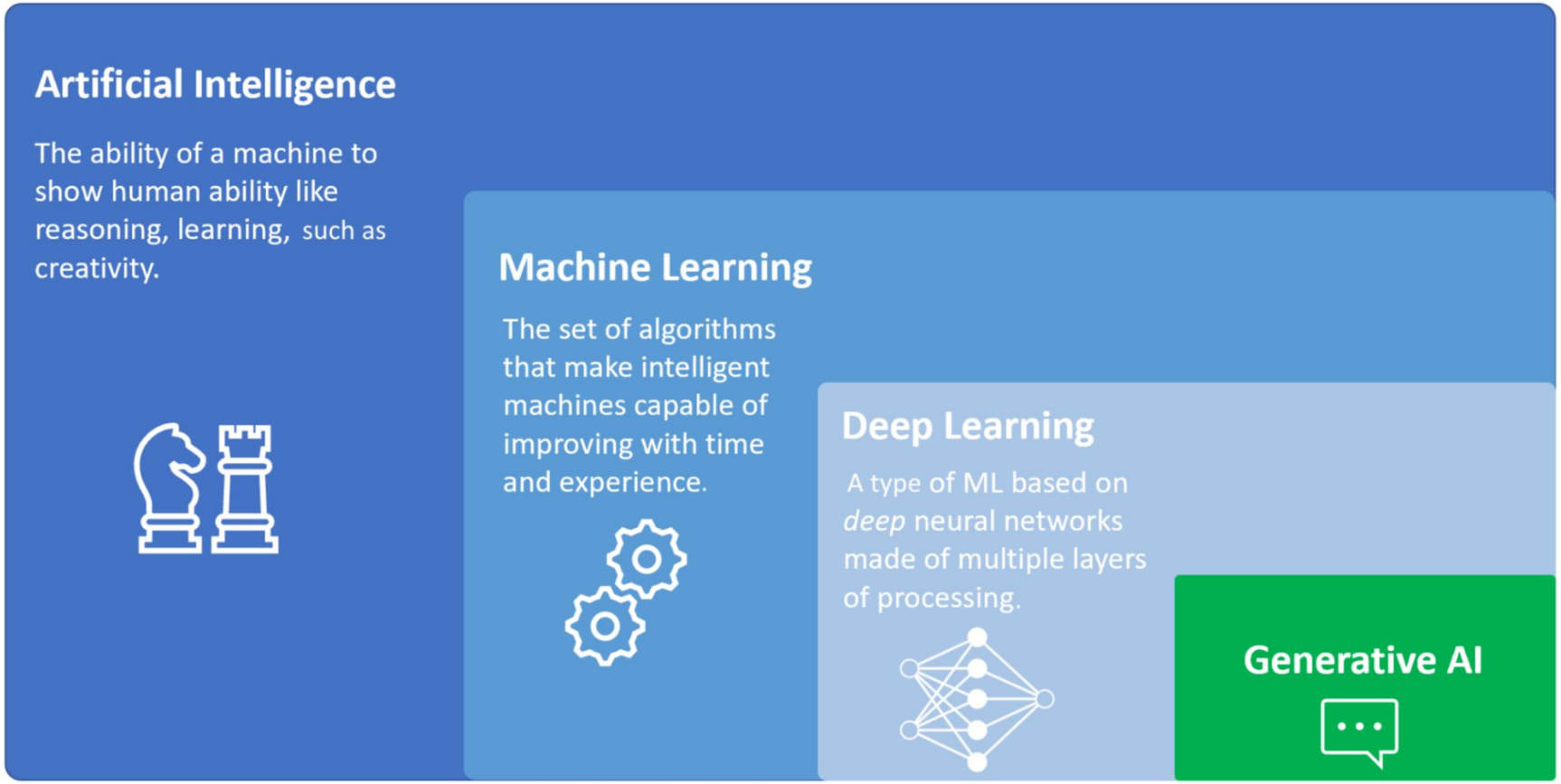


Input Layer

Hidden Layer

Output Layer

AI ⊃ ML ⊃ DL ⊃ GenAI

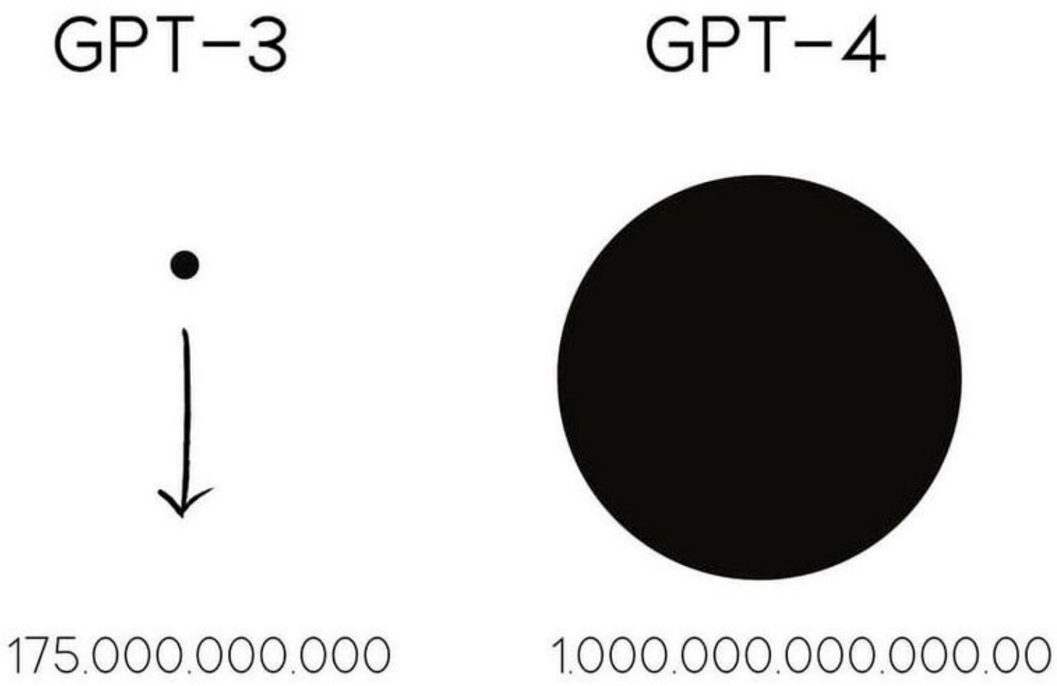


LLM

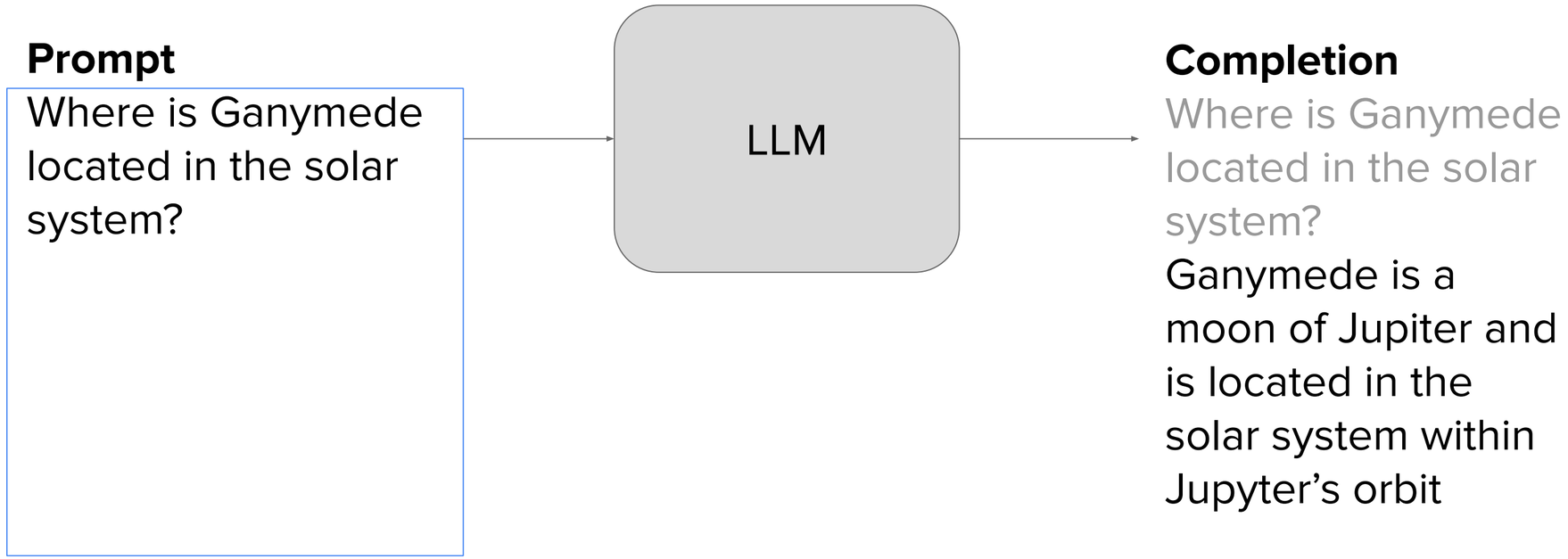
- **Large Language Model** (LLM) is a neural network with many parameters (typically billions of weights or more), trained on large quantities of unlabelled text using self-supervised learning
- A message is splitted in **tokens**
- Each token is translated in a number using an operation called **embeddings**
- LLM works by taking an input text and **repeatedly predicting** the next token or word

Size of GPT-4

- Around **1.76 trillion** parameters
- Neural network with **120** layers
- Process up to **25,000** words at once
- Estimated training cost is $200M using 10,000 Nvidia A100 GPU for 11 months

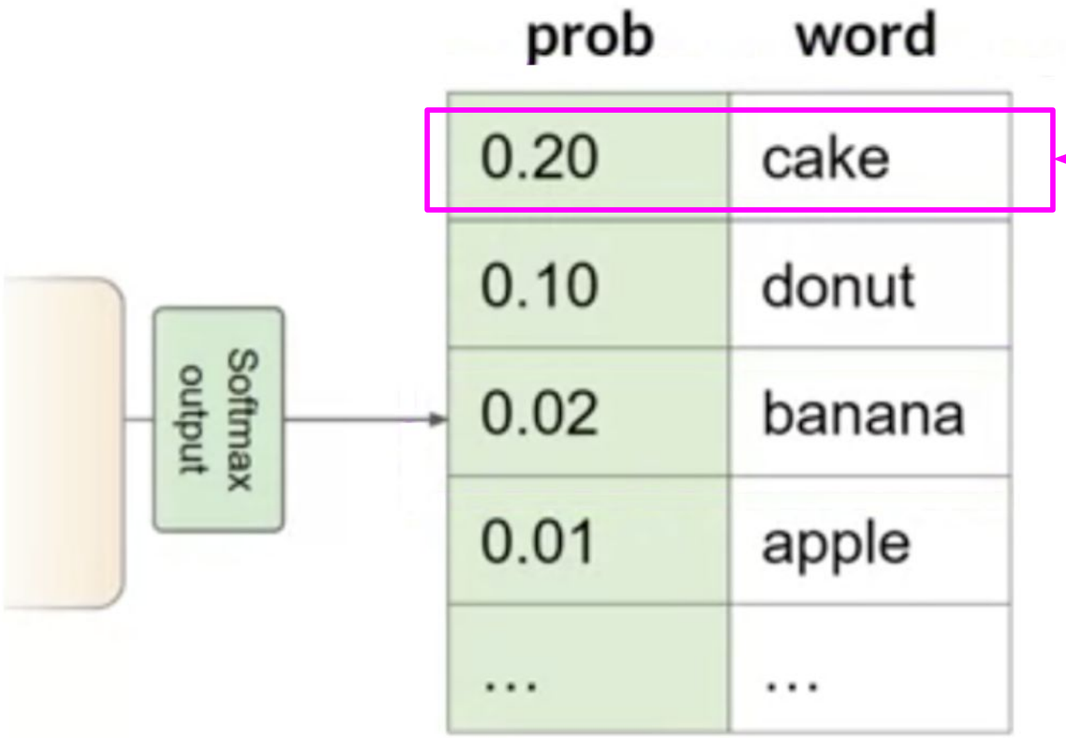


Prompt



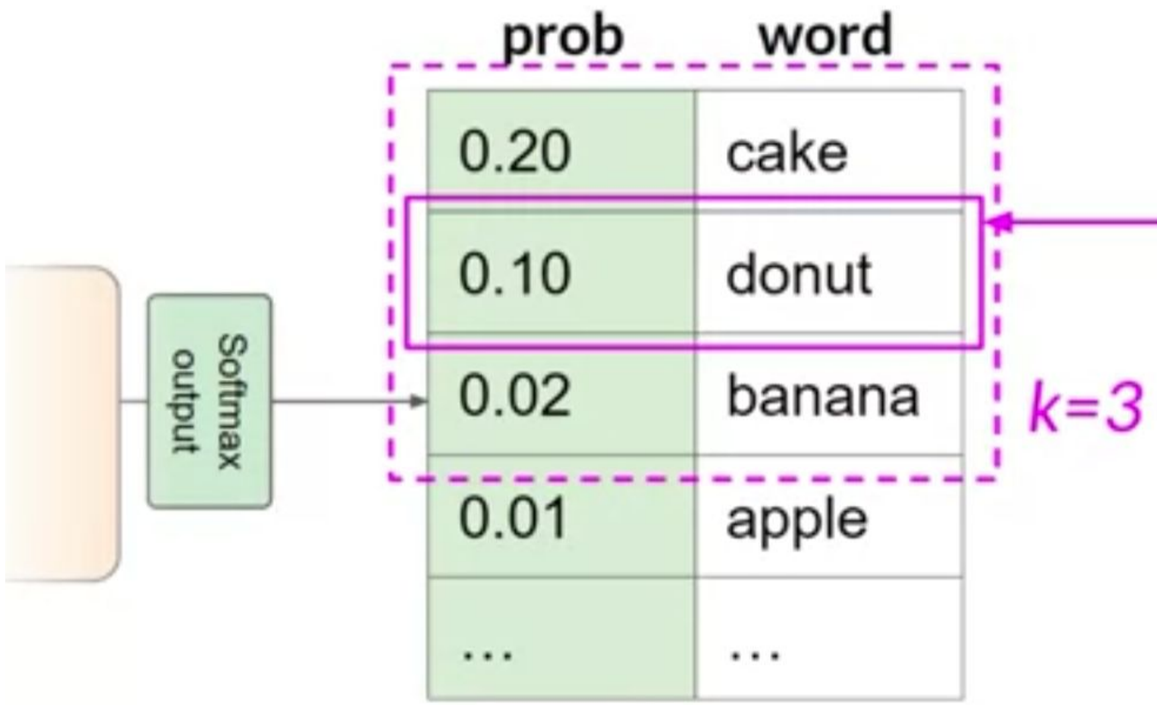
Context window: few thousand words

Predict the next word



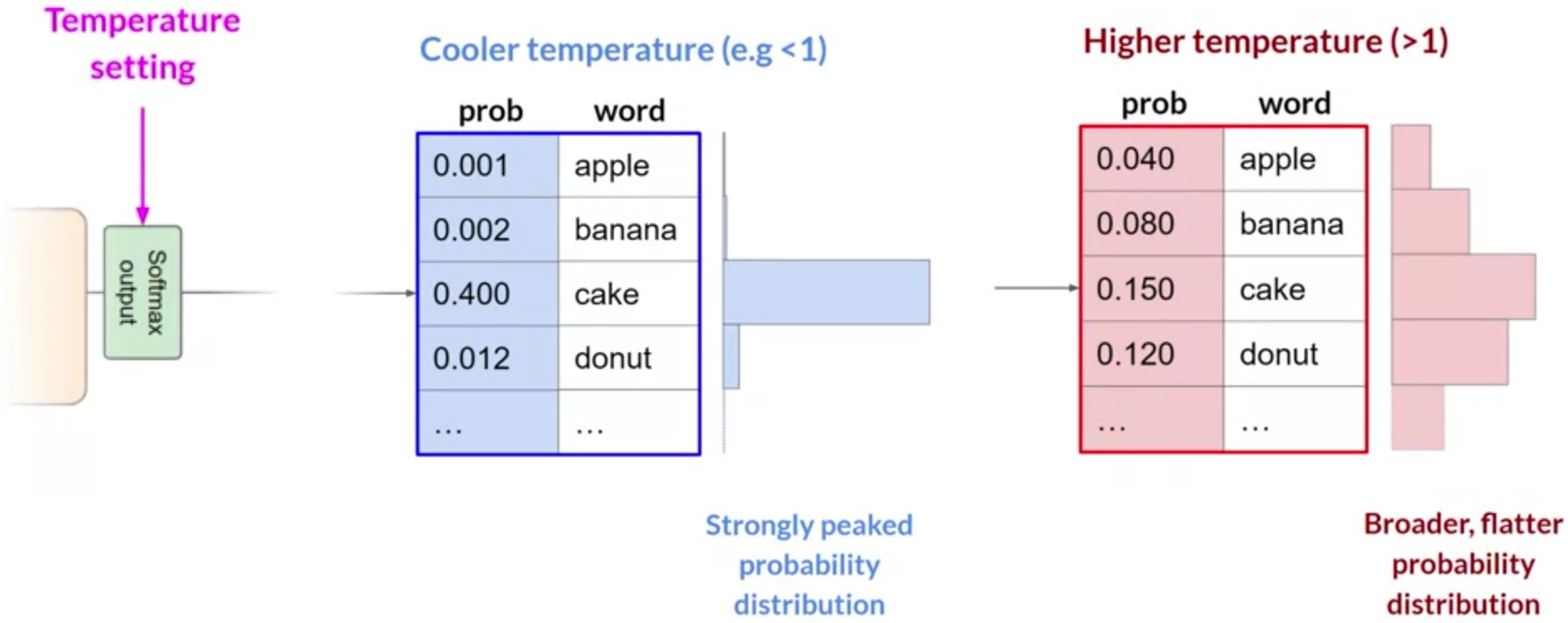
Choose the one with greatest probability (greedy algorithm)

Top-k



top-k: select an output from the top-k results after applying random-weighted strategy using the probabilities

Temperature



Ollama

- Ollama is a software for downloading and running (open source) LLMs
- Llama 2/3, Phi 3, Mistral, Gemma, and other models
- Simple interface:
  - ollama pull llama3.1
  - ollama run llama3.1

elastic

LLPhant

- LLPhant is a comprehensive open source Generative AI framework for PHP
- The goal is to offer an easy to use library to build GenAI applications in PHP
- LLM supported: OpenAI, Ollama, Mistral
- Vector databases: Elasticsearch, File, Memory, Milvus, Qdrant, Redis
- Started by Maxime Thoonsen and sponsored by Theodo

elastic

Example: LLPhant with Llama3

```php
use LLPhant\Chat\OllamaChat;
use LLPhant\OllamaConfig;

$config = new OllamaConfig();
$config->model = 'llama3.1';
$chat = new OllamaChat($config);

$response = $chat->generateText('What is the capital of Italy?');
// The capital city of Italy is Rome
printf("%s\n", $response);
```

elastic

Retrieval-Augmented Generation (RAG)

- **RAG** is a technique in natural language processing that combines information retrieval systems with **Large Language Models** (LLM) to generate more informed and accurate responses
- It is composed by the following parts:
  - **Retrieval-Augmented**
  - **Generation**

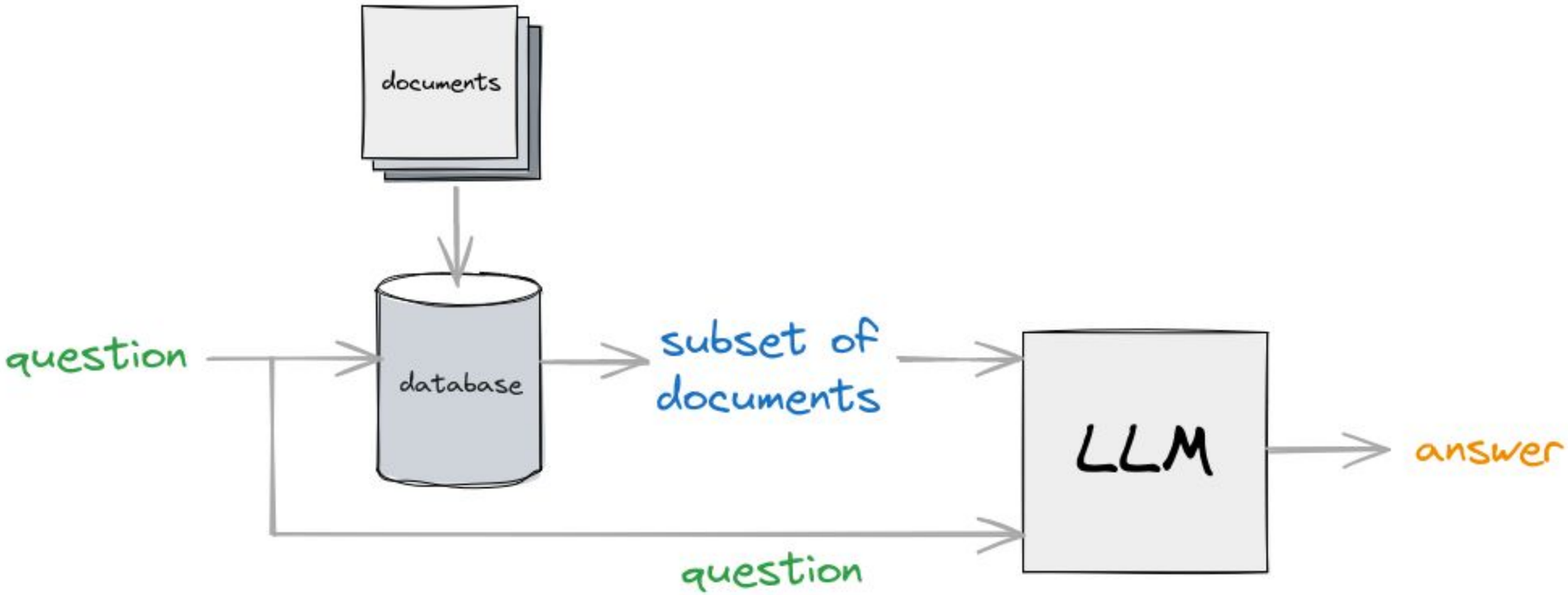
Generation

- LLMs are very powerful but have some limitations:
  - **No source** (potential hallucinations)
    - How can I verify the information coming from an LLM?
    - What sources has been used to generate the answer?
  - **Out of date**
    - An LLM is trained in a period of time
    - For update we need to retraining the model (very expensive)

Retrieval-Augmented

- We collect sets of private or public document
- We build a **retrieval system** (e.g. a database) to extract a subset of documents using a **question**
- Then we pass the **question + documents found** to an LLM as prompt with a context
- The LLM can give an answer using the updated documents

RAG architecture



elastic

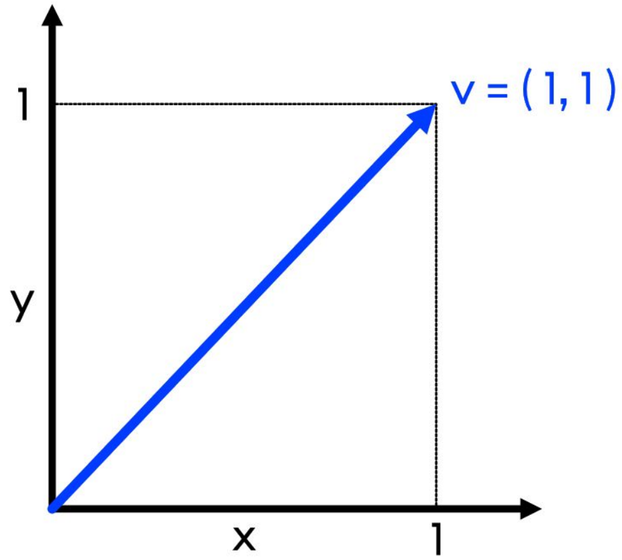
Retrieve documents from a question

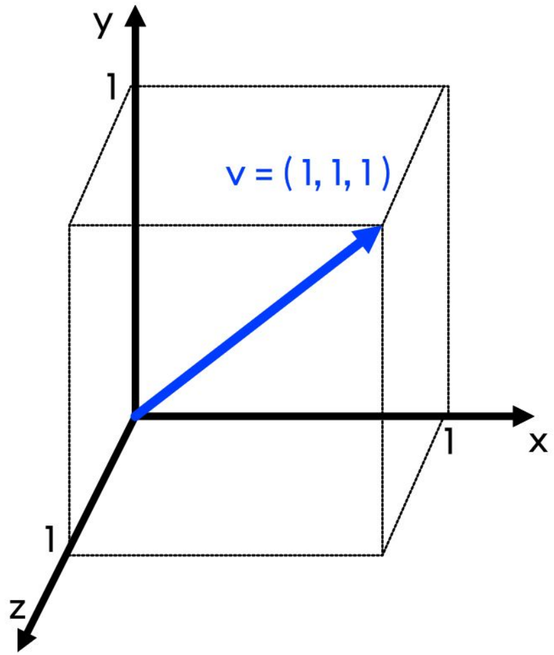
- How we can retrieve documents in a database using a question?
- We need to use **semantic search**
- One solution is to use a **vector database** (eg. Elasticsearch)
- A vector database is a system that uses **vectors** (set of numbers) to retrieve information

elastic

What is a vector?

- A vector is a set of numbers
- Example: a vector of 3 elements [2, 5, -10]
- A vector can be represented in a multi-dimensional space (eg. Llama3.1 uses 4096 dimensions)

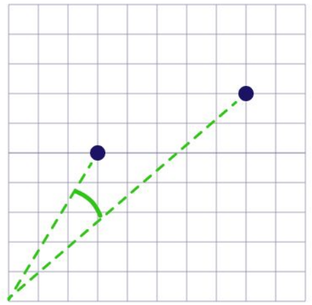




elastic

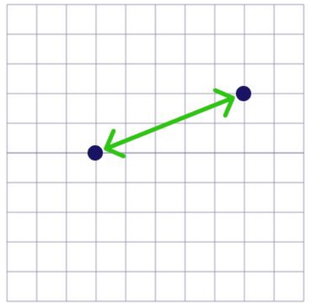
Similarity between two vectors

- Two vectors are (semantically) similar if they are close to each other
- We need to define a way to measure the similarity



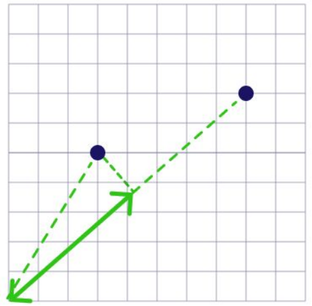
Cosine Distance

$$1 - \frac{A \cdot B}{||A|| \quad ||B||}$$



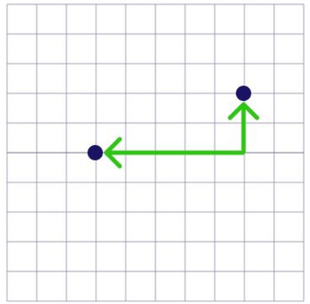
Squared Euclidean (L2 Squared)

$$\sum_{i=1}^{n} (x_i - y_i)^2$$



Dot Product

$$A \cdot B = \sum_{i=1}^{n} A_i B_i$$



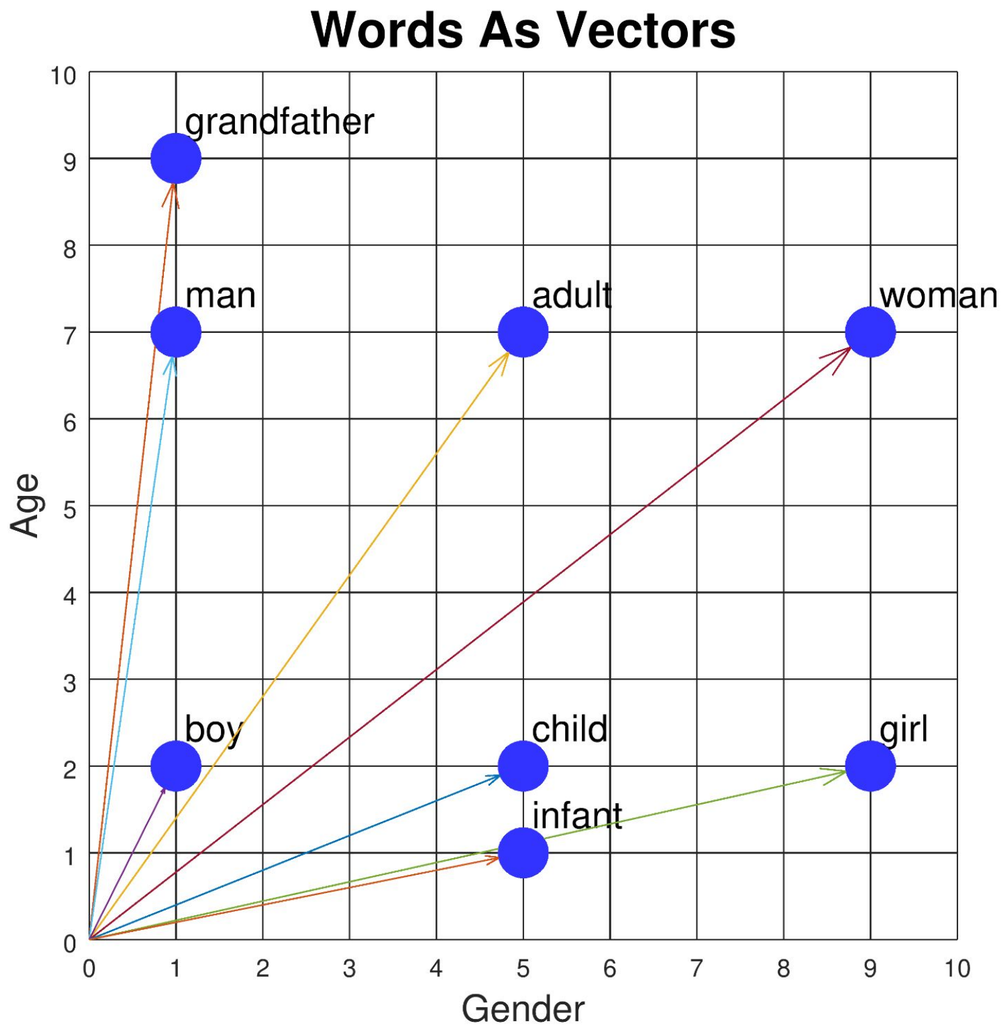
Manhattan (L1)

$$\sum_{i=1}^{n} |x_i - y_i|$$

elastic

Embedding

- Embedding is the translation of an input (document, image, sound, movie, etc) to a vector
- There are many techniques, using an LLM typically this is done by a neural network
- The goal is to group information that are semantically related to each other



elastic

Vector database + LLM

- The search query (***question***) is in natural language
- We use semantic search to retrieve top-n relevant documents (***context***)
- We send the following prompt to the LLM:
  - *Given the following **{context}** answer to the following **{question}***

elastic

Split the documents in chunk

- We need to store data in the vector database using chunk of information
- We cannot use big documents since we need to pass it in the context part of the prompt for an LLM that typically has a token limit (e.g. llama3.1 up to 128k)
- We need to split the documents in chunk (eg. number of words)

Build a RAG system in PHP:
Llama3 + Elasticsearch + LLPhant

Available on github: [ezimuel/php-llm-examples]

elastic

Steps

- Install Llama3.1
  - ollama pull llama3.1
- Ask to LLama3.1 “How many moons has Neptune?”
  - ollama run llama3.1
  - How many moons has Neptune?
  - The answer will be 14.
- Download php-llm-examples and install
  - git clone https://github.com/ezimuel/php-llm-examples.git
  - cd php-llm-examples
  - composer install
- Start Elasticsearch (and Kibana) at localhost
  - bin/start-local.sh

elastic

Steps (2)

- Index the PDF document in Elasticsearch (embedding)
  - php src/rag/fwdays/embedding.php
- Use the RAG of LLPhant to ask again “How many moons has Neptune?”
  - php src/rag/fwdays/qa.php
  - You should see an answer like:

    According to the text, Neptune currently has 16 known moons, but there are two new ones that have been discovered...

References

- What is retrieval-augmented generation? IBM research
- The Llama 3 Herd of Models, Meta
- Nathan Benaich, State of AI Report 2023, Air Street Capital
- Valentina Alto, Modern Generative AI with ChatGPT and OpenAI Models, Packt, 2023
- Ashish Vaswan et al., Attention Is All You Need, Proceedings of 31st Conference on Neural Information Processing Systems (NIPS 2017)
- Will Oremus, Google's AI passed a famous test — and showed how the test is broken, Washington Post, June 17, 2022
- Scott Mayer McKinney et al., International evaluation of an AI system for breast cancer screening, Nature, Vol. 577, 2 January 2020
- Albert Ziegler, John Berryman, A developer's guide to prompt engineering and LLMs, Github blog post
- Saurabh Mhatre, What Is The Relation Between Artificial And Biological Neuron?

Thanks!

More information: www.elastic.co