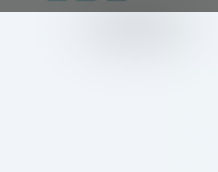
大模型系列 – 数据处理

# 向量数据库介绍



# 大模型业务全流程

## 数据 & 模型算法

### 2. 数据处理

开源数据 | 数据预处理 | 向量数据库

### 3. 模型算法

LLM模型架构 | 多模态一切皆Tokens

## 模型训练 & 微调

### 4. 模型训练

混合精度 | 梯度检查 | 梯度累积 | …

5. 分布式并行

训练集群稳定性

### 6. 模型微调

全参微调

低参微调

指令微调

## 模型验证 & 推理部署

### 7. 模型验证

下游任务 | 测评标准 | benchmark

### 8. 推理与智能体

量化压缩 | 推理加速 | 9. Agent 智能体

## 集群算力准备

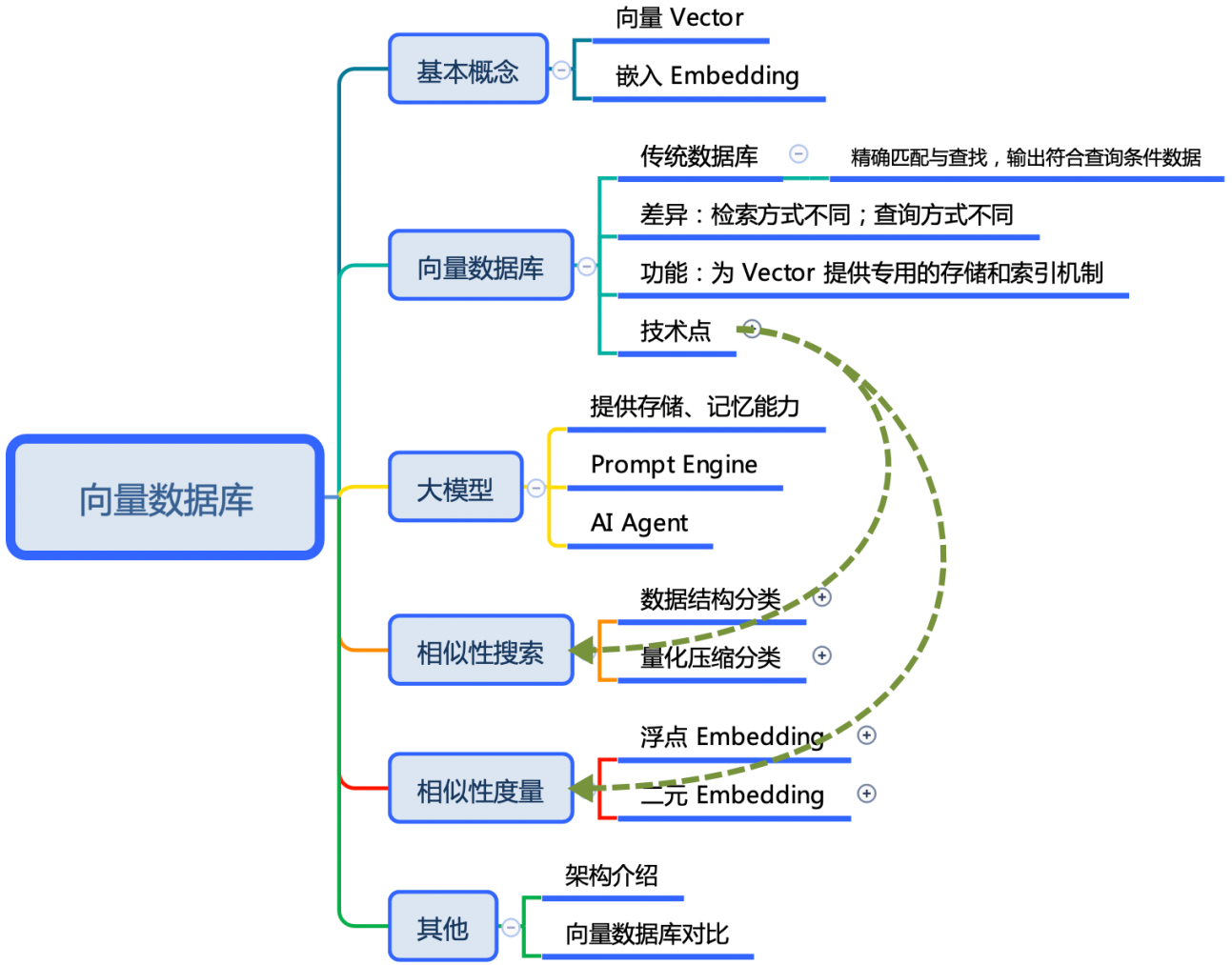
### 1. AI 集群建设

计算、存储、网络 | AI 集群机房建设 | AI 集群上线与运维

# 大模型系列 – 数据处理之向量数据库

- 具体内容
  - **向量与检索：**向量 Vector 的表示 -- Embedding 原理
  - **向量数据库：**向量数据库原理、功能、特点 -- Vector-DB 应用场景
  - **大模型关系：**向量数据库遇到大模型 – 大模型与 Vector-DB 应用场景
  - **相似性搜索：**K-Means 聚类 -- Faiss 算法 -- PQ 算法 -- IVF 算法 -- HNSW 算法
  - **相似性度量：**欧氏距离（L2） -- 内积（IP） -- 其他度量方式
  - **通用性架构：**通用 Vector-DB 架构 -- KDB 架构示例
  - **对比与小结：**业界向量数据库横向对比 -- Vector-DB 小结

# 大模型系列 – 数据处理之向量数据库



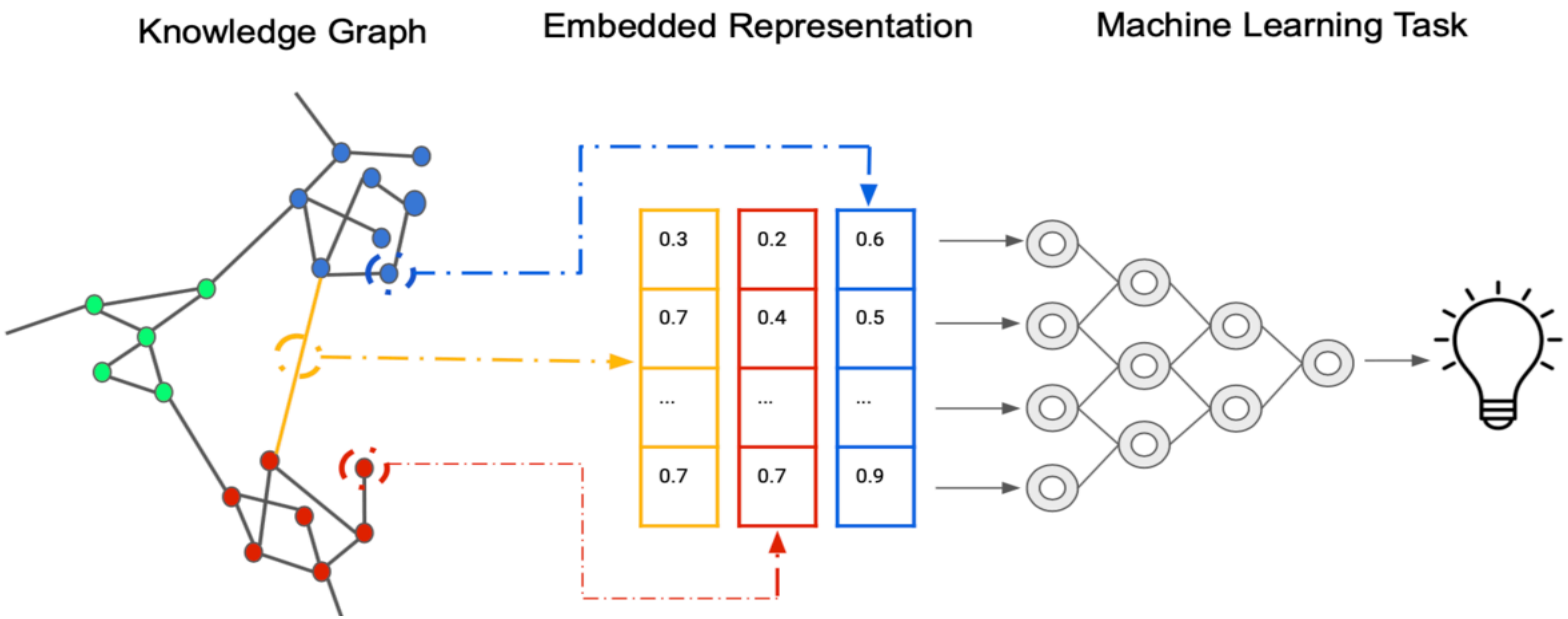
# 思考

1. 向量数据库 Vector-DB 对大模型的价值和意义是什么？
2. 向量数据库 Vector-DB 会不会随着大模型能力的提升而被代替？
3. 围绕着大模型配套软件开发工具，向量数据库处于什么样的角色定位？

# 1、背景介绍

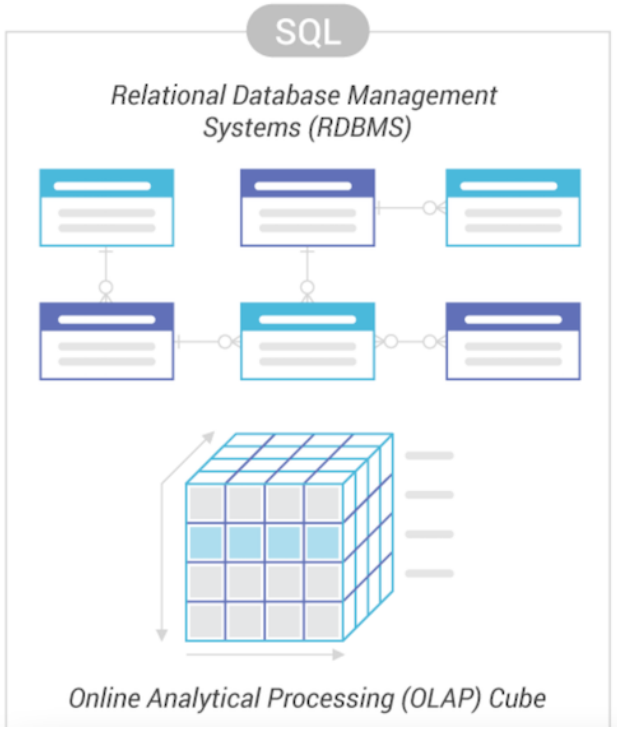
# 背景介绍 Background introduction

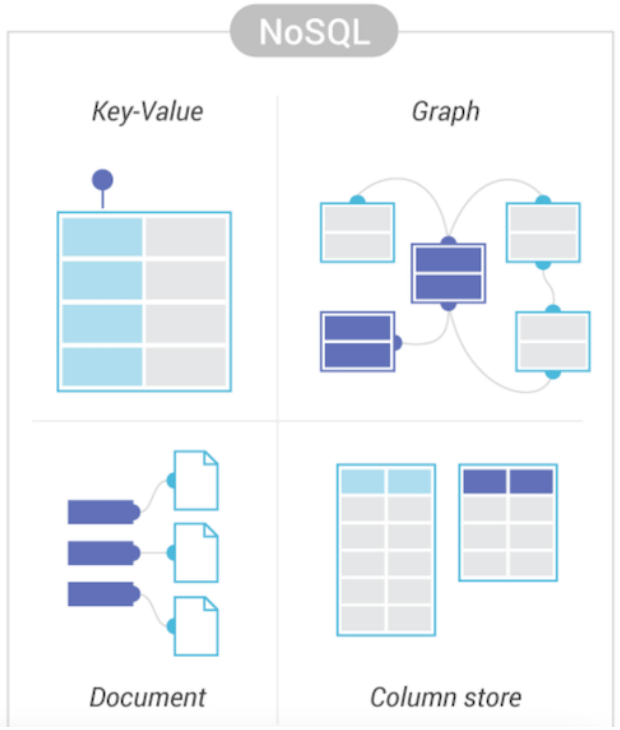
- 互联网非结构化数据以惊人速度增长，e.g. 文档、图像、视频和普通文本等形式。传统数据库针对结构化数据建立，处理非结构化数据变得越来越困难。
- 仅通过关键词分析、数据分类不足以完全表示挖掘和学习复杂数据所蕴含知识，特别在 AI 一切皆 Embedding Vector 时代。



# AI 与 Vector 的关系

- 移动互联网 JSON 支撑大规模灵活数据存储的通用格式，推动 MongoDB 流行；
- AI 时代向量 Vector 作为神经网络基本数据结构，Vector 也是大模型理解世界数据形式。



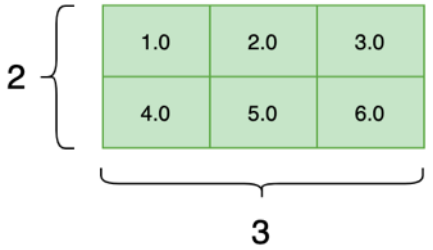



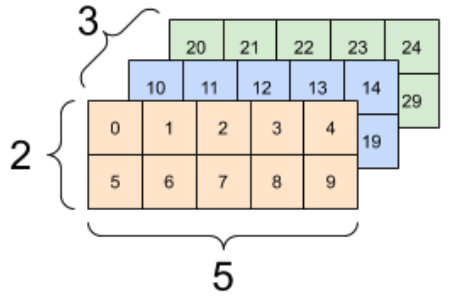
# 神经网络的基本组成回顾 Review of AI

## 基于数据流图（DAG）的计算框架

### 基本数据结构：Tensor 张量

- Tensor形状： [2, 3, 4, 5]
- 元素类型：int, float, string, etc.




### 基本运算单元：Operator 算子

- 由最基本的代数算子组成
- 根据深度学习结构组成复杂算子
- N个输入Tensor，M个输出Tensor

| | | |
|---|---|---|
| Add | Log | While |
| Sub | MatMul | Merge |
| Mul | Conv | BroadCast |
| Div | BatchNorm | Reduce |
| Relu | Loss | Map |
| Floor | Sigmoid | ..... |

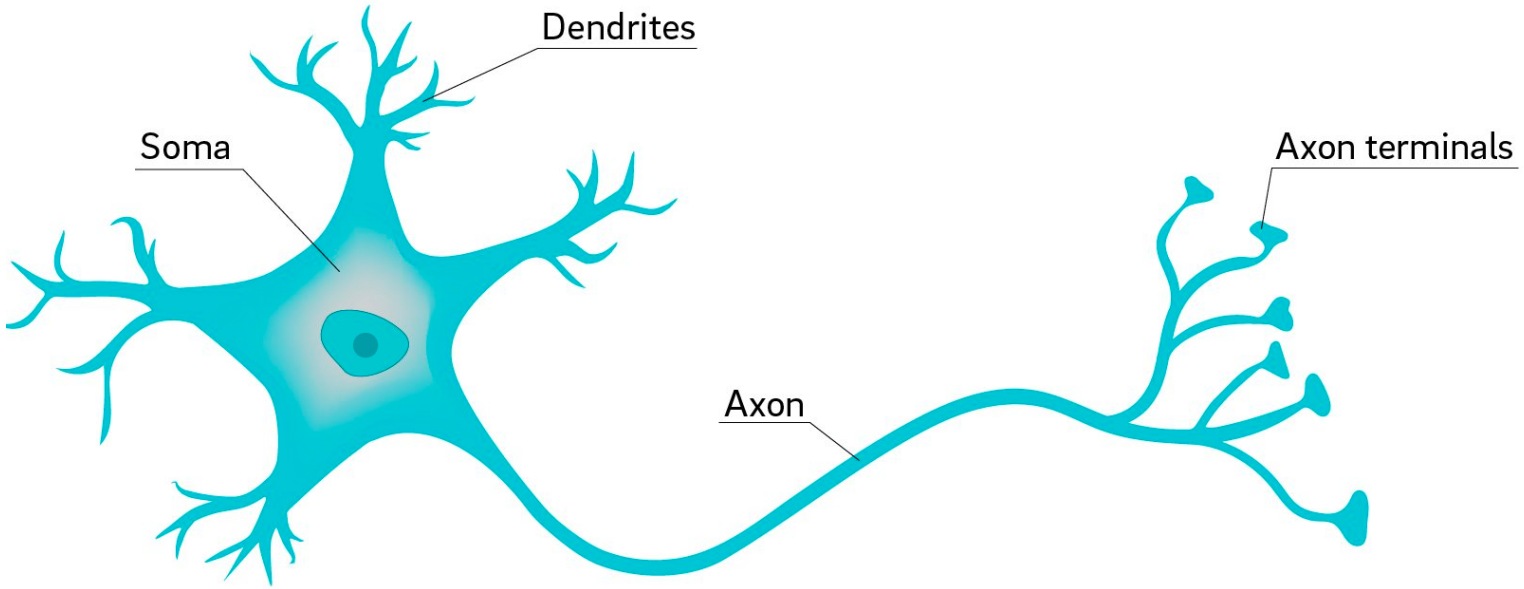
# 思考

1. 向量检索的核心在于相似性搜索（Similarity Search），在深入了解相似性搜索前，我们需要一起详细了解特征 Feature 和向量 Vector 的概念和原理。

# 2. Vector 向量

# 什么是向量 What is Vector

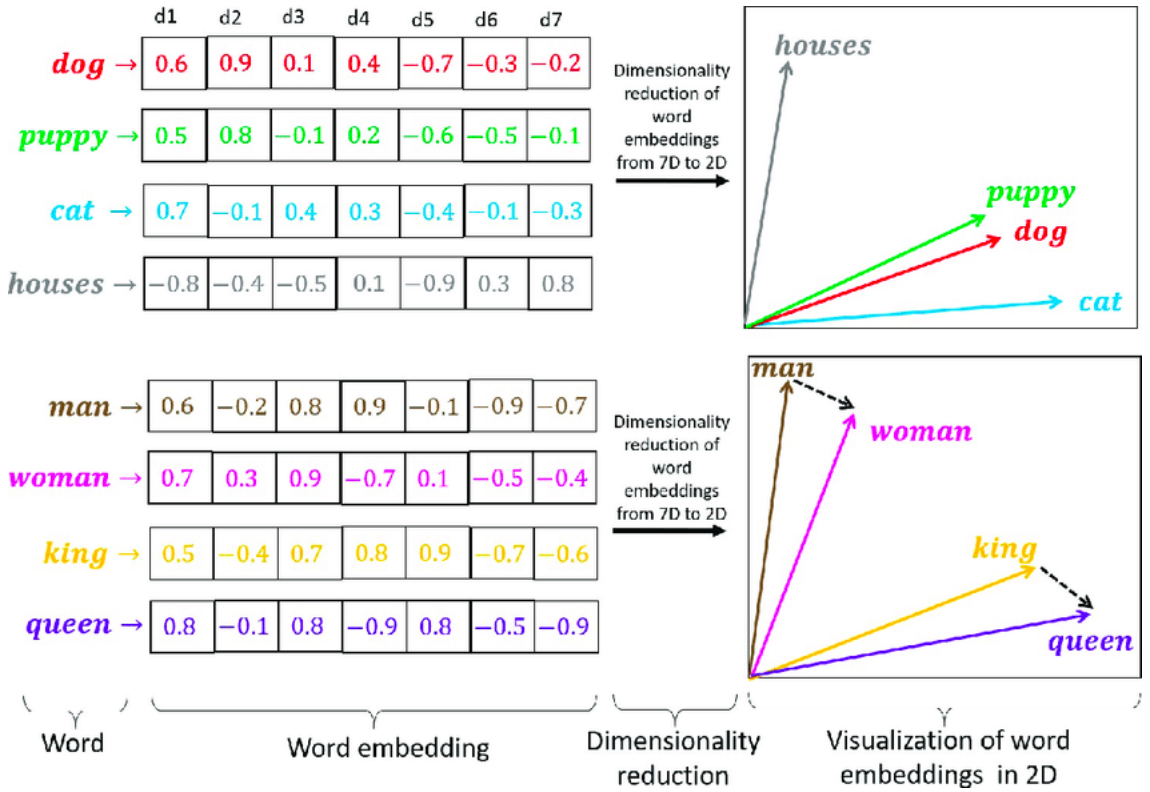
- AI 神经网络模型设计，在神经元激活后形成对输入数据表征，即人脑理解和学习的方式。
- AI 模型实际识别和理解非具体文字符号，而是神经网络对各类数据向量化 Vector 表示。



# 什么是向量 What is Vector

- 向量是什么？
- 向量从哪来？
- 向量怎么用？

- ✓ 向量是多维数学空间中的一个坐标点
- ✓ 向量来源于对这是世界数字化的抽象
- ✓ 向量用来搜索真实世界不同模态数据



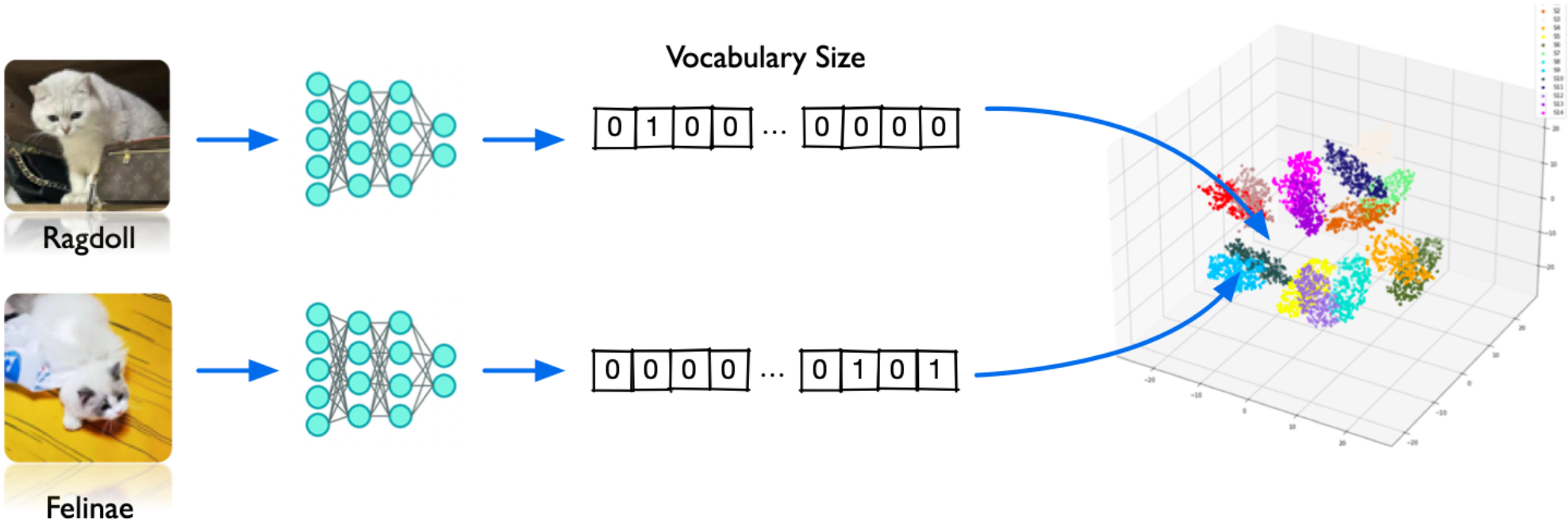
# 向量类型 Vector type

| 向量类型 | 详细介绍 |
| --- | --- |
| **图像向量** | 通过神经网络模型提取图像特征向量 Feature Map，特征向量 Feature Map 提取输入图像的高维信息，如颜色、形状、纹理等，最终可以用于图像识别、检索等任务； |
| **文本向量** | 通过词嵌入 Work Embedding 技术如 Word2Vec、BERT 等生成文本特征向量，文本特征向量包含文本语义高维信息，可以用于文本分类、情感分析等 NLP 任务； |
| **语音向量** | 通过声学模型 Acoustic Model 从音频信号中提取的音频的频谱特征向量，频谱特征向量提取声音的高维特性，如音调、节奏、音色等，可用于语音识别、声纹识别等任务； |

# 3. Embedding

# 什么是 Embedding

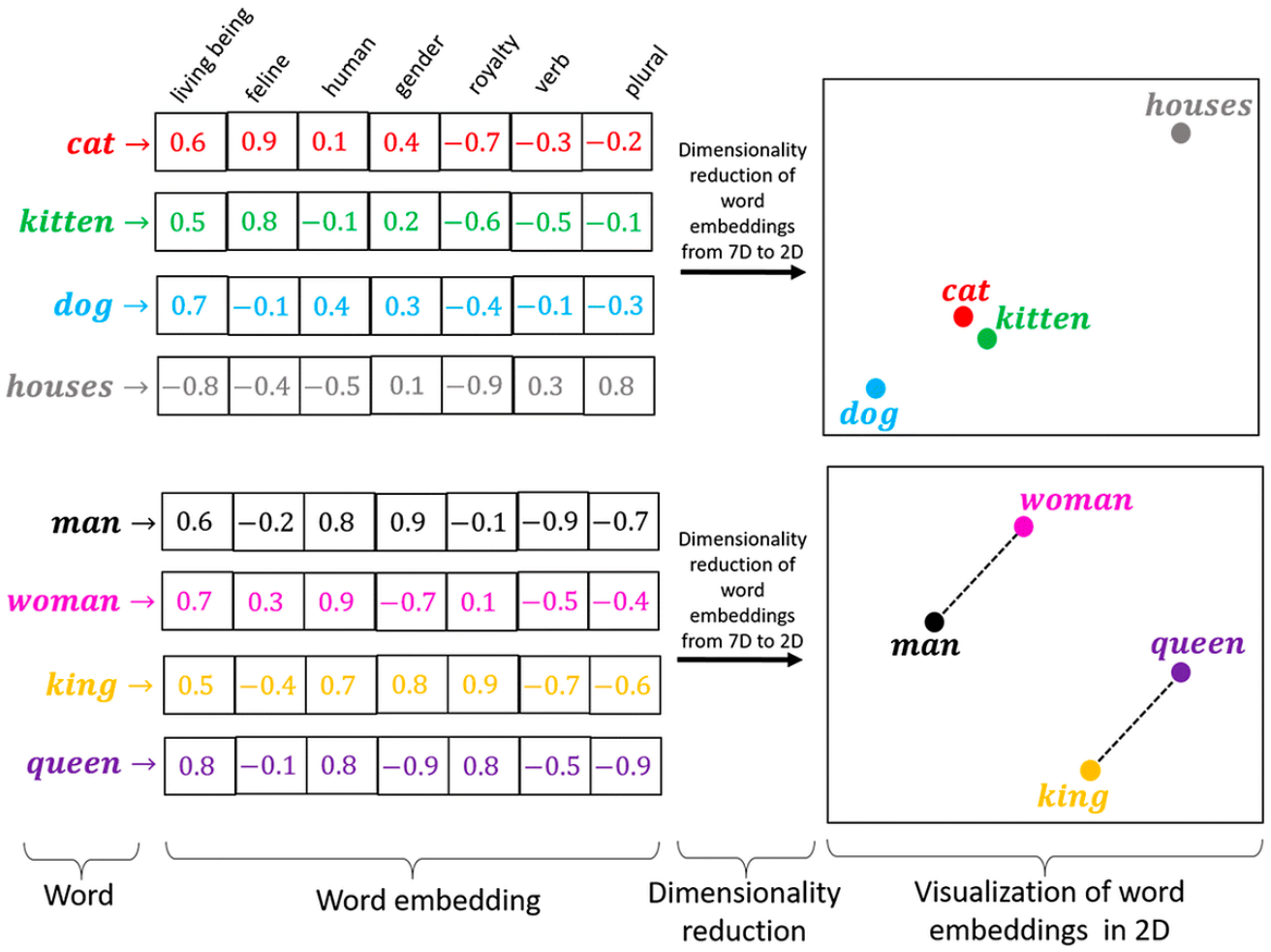
- **Embedding 介绍**：非结构化数据转换成向量的过程
- **Embedding 作用**：通过深度学习的训练，将真实世界离散数据，投影到高维数据空间上，通过数据在空间中间的距离体现真实世界的相似度



# 什么是 Vector Embedding

1. **Vector Embedding**：向量嵌入
   将非数值词语/符号等非结构化数据，编码成数值向量；

2. **Word Embedding**：词嵌入通过 NN 来学习，文本中词语作为NN 网络输入，输出对应词向量 Word Vector。词向量是一个数值向量，每个数值代表词语的某个特征。

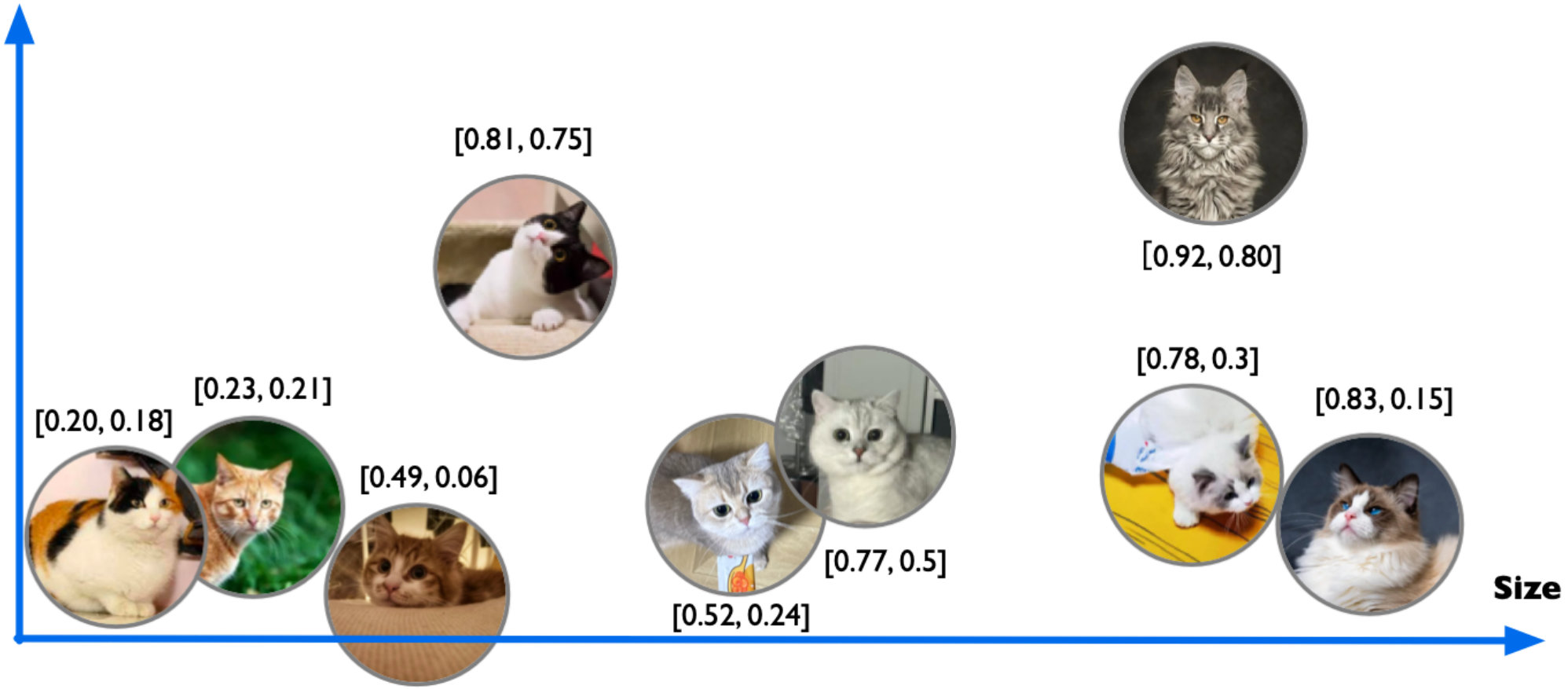


# Vector Embedding DEMO

- 人类通过不同事物间不同特征来识别种类，e.g.，如分辨不同种类小猫，可通过体型大小、毛发长度、鼻子长短、脸型等特征区分。
- e.g.，体型越大的猫在一维度坐标轴右边，得到一个体型特征与一维坐标从 0 到 1 对应数值。

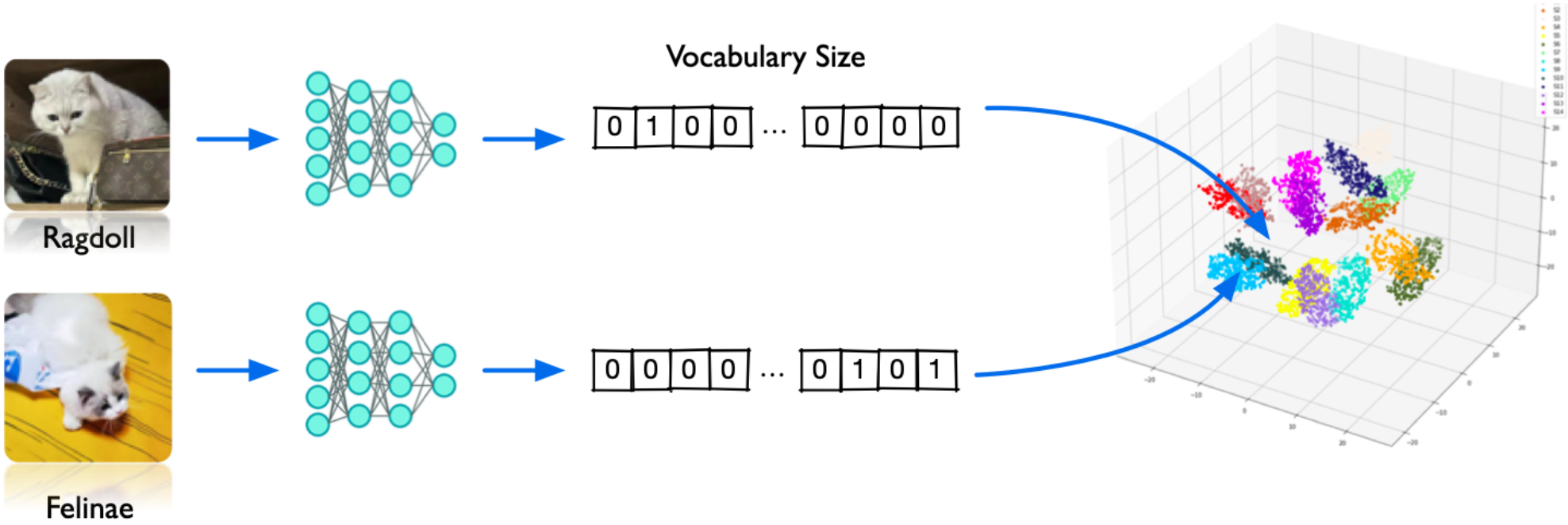
# Vector Embedding DEMO

- 单靠体型特征并不能充分区分不同种类的猫咪，e.g. 金渐层、银渐层的体型接近，难以通过体型直接区分。因此会增加其它特征，例如毛发颜色。



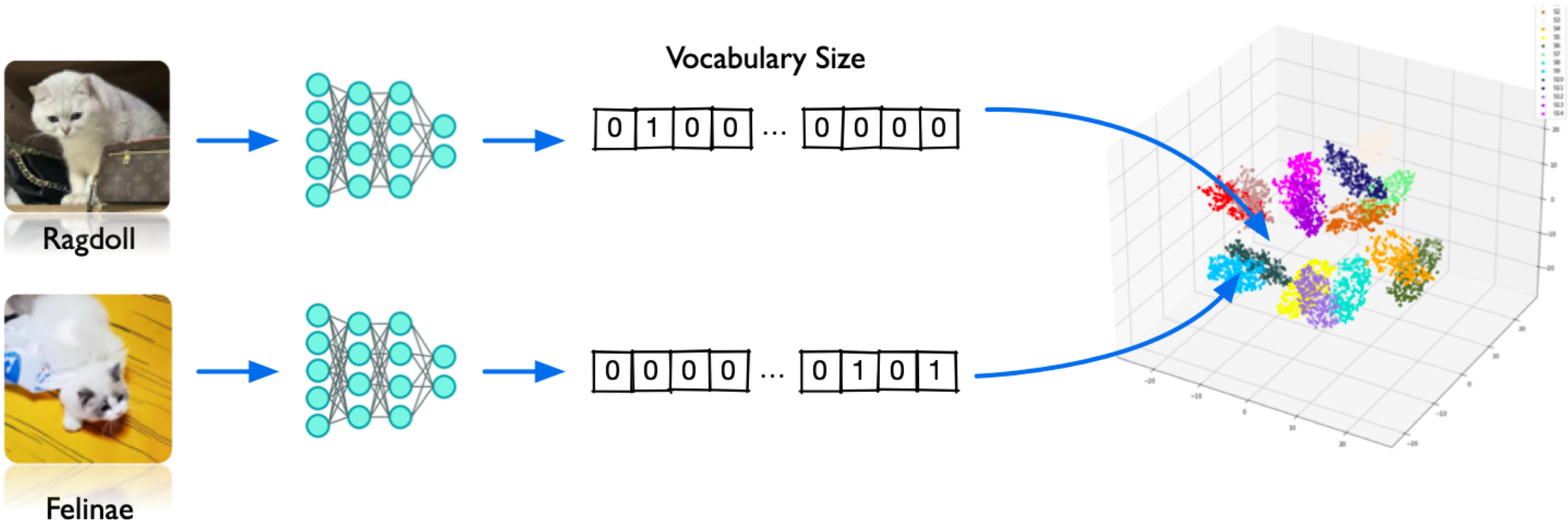
# 向量高维表示

- 只要特征维度足够多，能将所有猫区分开来，最后得到一个高维坐标系。
- 虽然难以想象高维坐标系，但是在向量数组中，只需要向数组中追加数值即可。



# 向量高维表示

- 通过 Vector Embedding 将 Ragdoll 和 Feline 表示为两个不同向量 [0100…000] 和 [0000…0100]
- 两个向量包含了 Ragdoll 和 Feline 特征，计算两向量距离来判断其相似度



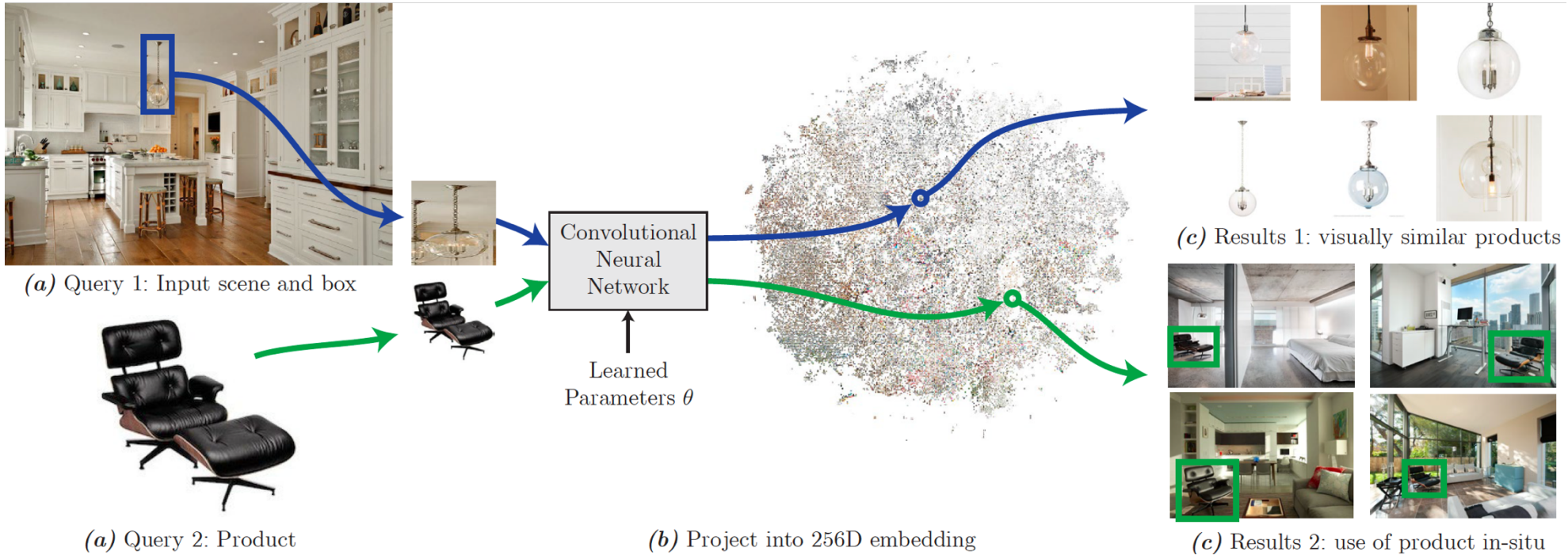
# 重点

- 向量维度越低，嵌入空间 Embedding Space 中特征表示就越紧凑，可能会影响下游任务或模型训练质量。

# 3. Summary

# 向量是 AI 理解世界的通用数据形式

- 无论游戏、网络、教育、医疗等，各行业领域中使用 AI 能力和场景越来越多。AI 框架基本组成为向量 + 算子，训练和推理时可以看做向量搜索/索引和向量计算过程。
- 因此可认为：**向量是 AI 理解世界的通用数据形式，未来向量将会成为 AI 的灵魂。**




*(a)* Query 1: Input scene and box

*(a)* Query 2: Product

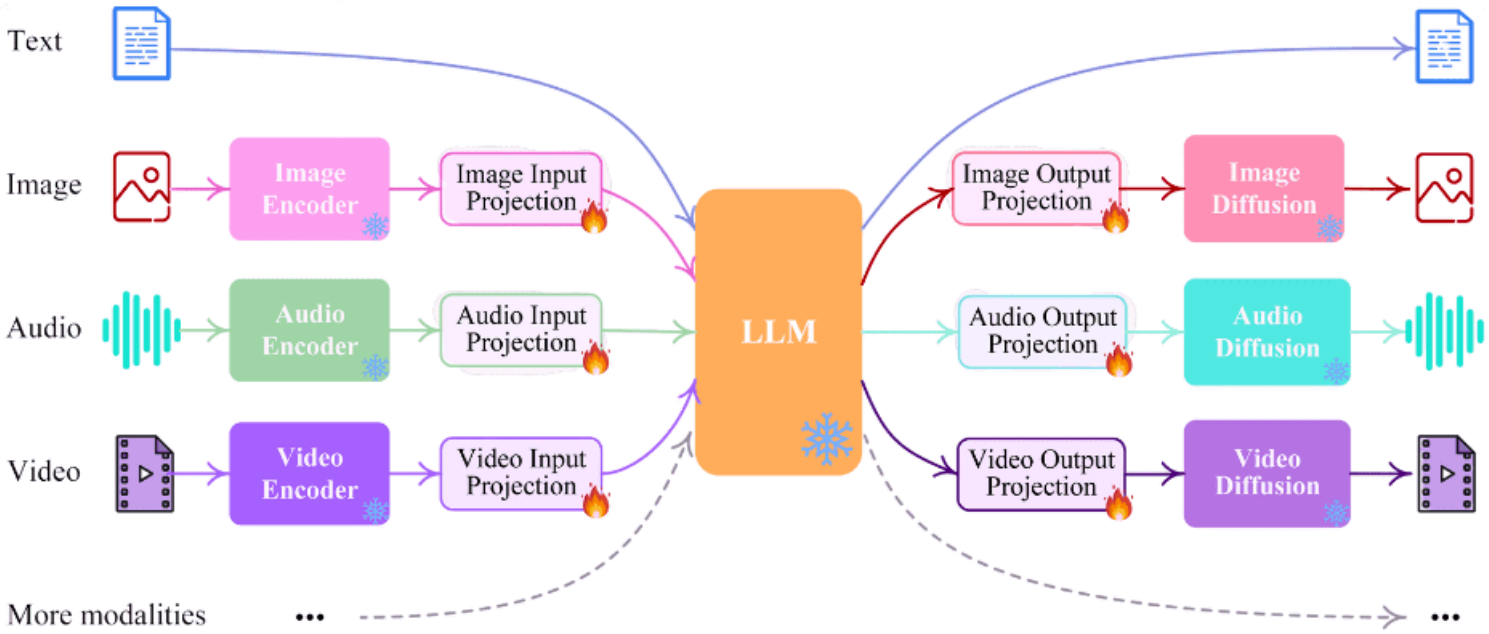
*(b)* Project into 256D embedding

*(c)* Results 1: visually similar products

*(c)* Results 2: use of product in-situ

# 多模态向量化是最终趋势

- LLM 大模型训练和推理过程，尽管大模型呈现出 E2E 文本输入输出，但实际模型接触和学习数据并非文本自身，而是向量化文本。
- 24/25 年通过对语音、图片、视频、文本数据进行向量化来训练多模态大模型，将是比较明确技术趋势，只有将一切数据向量化，才能让大模型更加聪明。



# 思考与小结

1. AI 经历从小模型到大模型变革，涉及数据量从 MB > TB > PB，模型所处理向量越来越大/多，这引发一个问题：**如何有效地存储和处理大规模向量数据？**
2. AI 应用不是简单精确匹配，而是进行复杂相似度检索，如何快速找出与给定向量 Vector 最相似向量，或查询一定范围内向量 Vector？

# Reference 引用&参考


1. Maximizing the Potential of LLMs: Using Vector Databases (ruxu.dev)
2. Maglott D， Ostell J， Pruitt KD， Tatusova T. Entrez Gene: gene-centered information at NCBI. Nucleic Acids Res. 2005 Jan 1;33 (Database issue):D54-8.
3. Wang Y， Xiao J， Suzek TO， Zhang J， Wang J， Zhou Z， Han L， Karapetyan K， Dracheva S， Shoemaker BA， Bolton E， Gindulyte A， Bryant SH. PubChem's BioAssay Database. Nucleic Acids Res. 2012 Jan;40 (Database issue):D400-12.
4. Torng W， Altman RB. 3D deep convolutional neural networks for amino acid environment similarity analysis. BMC Bioinformatics. 2017 Mar 16;18 (1):302.
5. Zheng S， Shao W， Chen L. UniVec: a database of gene expression vectors for PCA based gene similarity search. BMC Genomics. 2017 Dec 6;18 (Suppl 10):918.
6. Manning CD， Raghavan P， Schütze H. Introduction to Information Retrieval. Cambridge: Cambridge University Press， 2008.
7. Mikolov， Tomas， et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).
8. Andoni， Alexandr， and Piotr Indyk. "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions." Communications of the ACM 51.1 (2008): 117-122.
9. Jégou， Hervé， et al. "Product quantization for nearest neighbor search." IEEE transactions on pattern analysis and machine intelligence 33.1 (2010): 117-128.
10. Ge， Tiezheng， et al. "Optimized product quantization." IEEE transactions on pattern analysis and machine intelligence 36.4 (2013): 744-755.
11. Babenko， Artem， and Victor Lempitsky. "The inverted multi-index." IEEE transactions on pattern analysis and machine intelligence 37.6 (2014): 1247-1260.
12. https://www.bilibili.com/video/BV11a4y1c7SW
13. https://www.bilibili.com/video/BV1BM4y177Dk
14. https://www.pinecone.io/learn/vector-database/
15. https://github.com/guangzhengli/ChatFiles
16. https://github.com/guangzhengli/vectorhub
17. https://www.anthropic.com/index/100k-context-windows
18. https://js.langchain.com/docs/
19. https://github.com/weaviate/weaviate
20. https://redis.io/docs/interact/search-and-query/

# Reference 引用&参考


1. Muja M， Lowe DG. Fast approximate nearest neighbors with automatic algorithm configuration. VISAPP (1). 2009 Feb 4:331-40.
2. Jégou， Hervé， et al. "Product quantization for nearest neighbor search." IEEE transactions on pattern analysis and machine intelligence 33.1 (2011): 117-128.
3. Chen， Zhenjie， and Jingqi Yan. "Fast KNN search for big data with set compression tree and best bin first." 2016 2nd International Conference on Cloud Computing an d Internet of Things (CCIOT). IEEE， 2016.
4. Dehmamy， Nima， Albert-László Barabási， and Rose Yu. "Understanding the representation power of graph neural networks in learning graph topology." Advances in Neural Information Processing Systems 32 (2019).
5. Babenko A， Lempitsky V. The inverted multi-index. IEEE transactions on pattern analysis and machine intelligence. 2014 Jun 7;37 (6):1247-60.
6. Datar M， Immorlica N， Indyk P， Mirrokni VS. Locality-sensitive hashing scheme based on p-stable distributions. Proceedings of the twentieth annual symposium on Co mputational geometry. 2004 Jun 8:253-62.
7. https://www.pinecone.io/learn/series/faiss/locality-sensitive-hashing/
8. https://www.pinecone.io/learn/series/faiss/product-quantization/
9. https://www.pinecone.io/learn/series/faiss/locality-sensitive-hashing-random-projection/
10. https://www.youtube.com/watch?v=QvKMwLjdK-s&t=168s&ab_channel=JamesBriggs
11. https://www.pinecone.io/learn/series/faiss/faiss-tutorial/
12. https://www.youtube.com/watch?v=sKyvsdEv6rk&ab_channel=JamesBriggs
13. https://www.pinecone.io/learn/vector-similarity/
14. https://github.com/chroma-core/chroma
15. https://github.com/milvus-io/milvus
16. https://www.pinecone.io/
17. https://github.com/qdrant/qdrant
18. https://github.com/typesense/typesense

# Thank you

把AI系统带入每个开发者、每个家庭、每个组织，构建万物互联的智能世界

Bring AI System to every person, home and organization for a fully connected, intelligent world.



ZOMI

Course chenzomi12.github.io

GitHub github.com/chenzomi12/DeepLearningSystem