Vector Databases

Awase Khirni Syed

Sub-Title Size 32

DRAFT
ORIGINAL SERIES

Data

Role of Data in Artificial Intelligence

- Data reigns supreme, and computational advancements dictate the technological trends.
- Data can be classified as
  - Structured
  - Semi-structured
  - Unstructured
- The foundation of AI lies in data. Collecting and creating high-quality data is the most critical step and foundational to success of any artificial intelligence system.
- Diverse data sources contribute to training AI models, enabling them to learn patterns and make accurate predictions
- AI models learn from historical data and draw relevance based on historical data patterns. These data characteristics influence the outcome of the AI system’s quality, representativeness, and accessibility.
- Ethical handling of data in AI is crucial for ensuring fairness, inclusion and non-discrimination, as it significantly impacts the society at large.

Constraints of Modern LLM based systems

Constraints of Large Language Models

DRAFT
ORIGINALSERIES

1. Ambiguities and intricacies of natural language
   a. LLMs struggle with handling the nuances, context and multiple meanings inherent in human language
   b. Resolving ambiguity and understanding context remain an ongoing challenge
2. Costs and latency optimization
   a. Operating LLMs in production can be expensive due to computational resources and cloud costs
   b. Reducing latency while maintaining accuracy is crucial for real-time applications
3. Robot Model Re-training and monitoring
   a. LLMs require continuous re-training to adapt to evolving data and user needs
   b. Establishing robust monitoring systems ensures model performance and detects anomalies
4. Domain-specific limitations
   a. Fine-tuning LLMs for specific domains compromises their generality
   b. Some domains, like arithmetic operations or staying current with the latest information and may tend to pose some challenges
5. Operational overhead for local deployment
   a. Running LLMs locally involves complexities and operational overhead
   b. Setting up and maintaining software and infrastructure can be time-consuming
6. Limited Scalability Locally
   a. Local deployment lacks the scalability of cloud-based solutions
   b. Upscaling or downscaling on demand is challenging when running LLMs locally.

Data Constraints of Modern LLM based systems

Constraints of Large Language Models

DRAFT
ORIGINALSERIES

1. Technological constraints
   a. Large scale LLMs present numerous technical challenges in terms of computational power and resources.
   b. The training of large language models require substantial processing and storage capacity, thereby limiting our ability to process large amount of information to derive patterns.
   c. LLMs use tokens, units of text or code to process and generate language, but there is a limit on the number of data tokens available for data sets.
2. .Data Quality Assurance
   a. The quality of data used to build data sets that feed into the AI system define the accuracy of the system
   b. Limited availability of high-quality or fine grained data results in weak predictions
   c. Data de-duplication is a necessary step to improve the quality of feeder data input to the AI system
3. Temporal Context and Dynamic Information
   a. LLMs lack a notion of temporal context. They struggle with handling dynamic information such as current weather, current stock price etc. Existing data needs to be enriched to address some of these limitations
4. Multimodal Learning Challenges
   a. While LLMs are progressing from text-only models to multi-modal models (based on images, videos, audio, etc.), they are not yet advanced enough to learn dynamically from different modalities. Hence achieving this objective of self-learning from different modalities accurately still remains a challenge.

Vector

Definition

DRAFT
ORIGINALSERIES

- A vector is fundamental mathematical concept. It’s essentially an array of numbers, each representing a specific dimension. For instance, in a 2D vector, we have two components (x and y), while in a 3D vector, we have three components (x, y and z).
- Vectors are widely used in various fields, including physics, computer graphics and machine learning
- Modern CPUs and GPUs are optimized for vector operations in machine learning algorithms.

- An array of numbers (weights)
- Representation of similarity
- Used for similarity searches
- Not unique to LLMs or AI
- Vector DBs make use of Vector Index’s

Vector Database

Definition

- It is designed to store, manage and index massive quantities of high dimensional vector data efficiently. Unlike traditional relational databases with rows and columns, data points in a vector database are represented by vectors with a fixed number of dimensions, clustered based on similarity.
- They enable efficient storage and retrieval of high-dimensional vector data, supporting AI-driven applications and handling the growing volume of unstructured information
- Vector Data Representation
  - Data points are represented by vectors. These vectors can represent complex objects like words, images, videos, or audio
  - Some examples include
    - Text – words, paragraphs and entire documents represented as vectors
    - Images : Pixels combined into high-dimensional vectors for each image
    - Speech/Audio: Sound waves converted into numerical data and represented as vectors
- Vector Embeddings
  - They represent vectors in a continuous multi-dimensional space
  - These embeddings are generated by specialized models that convert raw vector data into an embedding
  - The goal is to handle millions of vectors efficiently, similar to similarity search

Vector Database

Definition

DRAFT
ORIGINALSERIES

- Application that involve large language models, generative AI, and semantic search rely on vector embeddings a type of semantic data representation.
- Vector embeddings are generated by AI models and have a large number of attributes or features, making their representation challenging to manage. In the context of AI and machine learning, these features represent different dimensions of the data that are essential for understanding patterns, relationships, and underlying structures. They offer optimized storage, indexing and querying capabilities for embeddings.
- It is a numerical representation of the data that captures the semantic relationships and similarities. Let's image representing data points as points in an n-dimensional space, where similar data points cluster together.
- Analogy: picture a bowl of various types of seeds/beans. We want to sort them by their size, type and color. Let's apply a mathematical representation to their positions. Each bean gets assigned a value based on its attributes. This process aligns with what a vector embedding does- assigning a mathematical representation to data points.
- Unstructured Data Handling
  - Traditional relational databases/Scalar databases are suitable for structured and semi-structured datasets
  - Vector databases excel at managing unstructured datasets through high-dimensional vector embeddings. They are ideal for scenarios where unstructured data, such as social media posts, images, videos and audio clips needs to be indexed and searched.
- They empower AI models to learn and grow.

Applications of Vector Embeddings

Transform raw data into powerful mathematical equations, enabling machines to learn, generalize and make informed decisions

- Text Analysis: Vector embeddings allow us to compare and analyze text documents. Similar documents end up close to each other in the embedded space
- Recommendation Systems: By embedding user preferences and item features, recommendation systems can suggest relevant products or content.
- Search Algorithms: Vector embeddings enhance search relevance by capturing semantic relationships between queries and documents
- Voice assistants and language translators: these systems rely on embeddings to understand context and provide accurate responses.

How are vectors used in machine learning?

Essential building blocks

DRAFT
ORIGINALSERIES

- Feature representation
  - In machine learning, data is often represented as feature vectors. Each data point (e.g., an image, text or audio clip) is transformed into a vector
  - For instance
    - In natural language processing (NLP), sentences are converted into word vectors (such as Word2Vec or GloVe embeddings)
    - Images are represented as pixel intensity vectors or deep learning features (e.g., from convolutional neural networks)
    - Audio signals become spectrogram vectors
- Distance Metrics and Similarity:
  - Vectors enable us to measure distances and similarities between data points using Euclidean distance (L2 norm) and cosine similarity metrics, which are widely used to compare vectors in recommender systems, clustering operations and anomaly detection
- Classification and Regression:
  - Machine learning models operate on feature vectors
  - Linear classifiers (e.g. logistic regression, support vector machines) learn decision boundaries in vector space.
  - Regression models predict continuous values based on input vectors.
  - For instance: predicting a product purchase based on a consumer's age, income and education level. Alternatively, predicting house prices based on features like square footage, location and number of bedrooms. Finally, predicting living expense computation based on existing lifestyle for monitoring money laundering against suspects.

How are vectors used in machine learning?

Essential building blocks

DRAFT
ORIGINALSERIES

- Embeddings and Representation Learning
  - They are used to capture semantic relationships and context. For e.g. word embeddings (e.g. Word2Vec,FastText) represent words as dense vectors.
  - Image embeddings (from pre-trained Convolution Neural Networks (CNN)) encode visual features. They model performance by capturing latent structures
- Neural Networks and Deep Learning:
  - They process vectors as input features and hidden layers. These layers in deep learning models transform input vectors into higher-level representations.
  - Convolution Neural Networks (CNN) operate on image patches (vectors), while Recurrent Neural Networks (RNN) process sequential data (e.g., text, time series data)
- Dimensionality Reduction:
  - High-dimensional data can be challenging to work with. By adopting techniques like Principal Component Analysis (PCA) to reduce dimensionality while preserving essential information. i.e. projection of data onto a lower- dimensional subspace, we retain critical features.

Regularization

Terminology and Definitions

- A technique that helps preventing overfitting and ensures better generalization of models.
- Overfitting: It occurs when a model becomes too specialized in the training data and fails to perform well on unseen data. It memorizes noise rather than learning meaningful patterns
- Underfitting: it occurs when a model is too simplistic and cannot capture essential patterns in the dataset
- Bias: It refers to errors when a model does not fit the real-world data well. High bias results in poor performance
- Variance: It represents errors due to prediction on unseen data. High variance occurs when the model learns noise from the training data

- Regularization Techniques
  - L1 Regularization – it adds the absolute value of coefficients as a penalty term to the loss function and encourages sparsity, aiding feature selection
  - L2 Regularization: It uses the square root of the sum of squared coefficients as a penalty term. It shrinks all coefficients to prevent overfitting
  - Elastic Net Regularization: It combines both L1 and L2 regularization. It balances feature selection and coefficient shrinkage
- Benefits of Regularization
  - Prevent Overfitting: by penalizing large coefficients, regularization discourages complex models that fit noise
  - Improve generalization: Regularized models perform better on unseen data
  - Reduces complexity: it simplifies the model by ignoring less important features

Different between L1 and L2 Norm

Comparative analysis

- L1 Norm (Manhattan Norm/Manhattan distance)
  - It is based on absolute values of the differences between the coordinates of two vector points.
  - It is computed by summing up the absolute values of each entry in the vector.
  - Used for sparse solutions, where L1 regularization encourages some weights to become exactly zero, leading to a feature selection
  - In addition to this, this approach is less sensitive to outliers.
  - L1 norm tends to produce more sparse solutions, where many weights become zero.
- L2 Norm (Euclidean Norm/Euclidean distance):
  - It is based on the square root of the sum of squares of the entities in the vector
  - It is used for gradient based optimization by penalizing large weights, promoting smoothing.
  - L2 norm shrinks all weights but not necessarily to zero.
  - L2 norm is sensitive to outliers and is commonly used due to its mathematical properties and smoothness.

Criteria for Selection of Regularization Technique

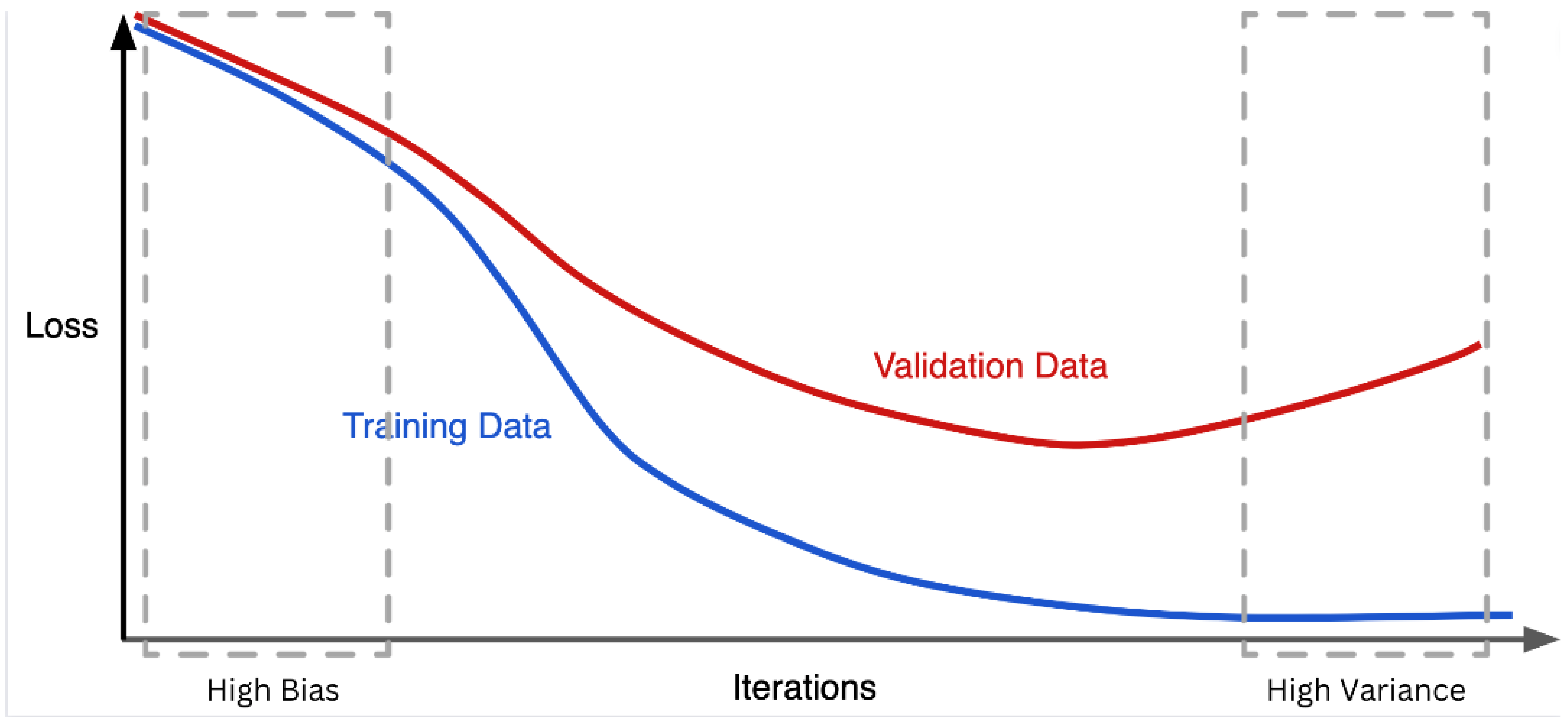
Factors for selecting the right technique

- Regularization is a crucial technique in machine learning to prevent overfitting and enhance model performance. Regularization aims to control model complexity by adding a penalty term to the loss function

Regularization=Loss Function + Penalty

- When a model fits too closely to the training data, capturing all its details and noise. A model may not generalize well to unseen data.
- **The generalization curve illustrates the phenomena where training loss decreases, validation loss eventually starts increasing due to increased model complexity.**
- **High variance models (overfitting) learn noise, while high bias models (underfitting) fail to capture essential patterns**

- The following factors define the criteria
  - Data characteristics: if you suspect some features are irrelevant, try L1 regularization (Lasso)
  - Feature Count: for many features, L2 regularization (Ridge) or Elastic Net may be better
  - Trade-off: Elastic Net balances L1 and L2 penalties
  - Experiment: Try different techniques and evaluate their performance using cross- validation



Regularization Rate

Model validation metrics

DRAFT
ORIGINALSERIES

- It is controlled by a scalar value called **λ** (lambda). It determines the strength of regularization penalty.
- When **λ** (lambda) is large, the regularization effect is stronger, leading o simpler models
- When **λ** (lambda) is small (or zero), the regularization effect diminishes, allowing the model to fit the training data more closely.
- **High λ**: Simpler model, risk of underfitting (won't learn enough from training data).
- **Low λ**: More complex model, risk of overfitting (won't generalize well to new data).
- **Ideal λ**: Data-dependent, requires tuning.
- Cross-validation helps find an appropriate **λ** evaluating model performance on validation data.
- **Note**: Setting λ to zero removes regularization completely, posing the highest overfitting risk

Purpose

Why vector databases matter for LLMs

LLMs have their limitations. They are not always adept at handling specific queries or understanding nuanced user requests. Vector databases store and search for embedding in high-dimensional space.

- They store domain texts and past user queries as numerical embeddings. When prompted with a new query, it encodes it into a vector and performs a similarity search. It finds similar vectors in its database and guides you to the right answer
- LLMs + Vector database: when combined together, they provide lightning-fast retrieval of relevant information using LLMs natural language handles

Usage of Vector Database

How to use Vector Databases in LLM Applications

DRAFT
ORIGINALSERIES

1. Encode Embeddings:

- Understand how to encode data into embeddings using embedding models (like Word2Vec or BERT)

2. Architect LLM App:

- Integrate vector databases into your LLM application architecture for efficient data retrieval

3. Code with TensorFlow:

- Dive into coding LLM or Generative AI Applications using vector databases and TensorFlow

- Example real-world applications
  - Chatbot App: Build a chatbot that understands user queries better than ever
  - Image Generator App: Create an app that generates images based on user description
  - Movie Recommendation App: helps users discover their next favorite movie

List of Vector Databases

Factors for selecting the right technique

Name	Description	Use Cases
Pinecone	A managed cloud-native vector database with a straightforward API and no infrastructure requirements	Ideal for application like e-commerce suggestions, image search and semantic search
Milvus	An open source vector database designed for vector embedding, efficient similarity search, and AI applications	
Weaviate	A vector database that support semantic search	Well-suited for applications involving natural language understanding and context-based queries
Deep Lake	A vector database designed for AI and machine learning	Useful for handling high-dimensional vectors in various applications
Qdrant	Vector database with similarity search capabilities	Suitable for applications requiring efficient similarity matching
Elasticsearch	While not exclusively a vector database, but it supports similarity search	Widely used for full-text search and analytics
Vespa	An open-source big data serving engine that includes vector search capabilities	Suitable for large-scale applications requiring real-time data retrieval
Chroma	Another vector database solution	Useful for applications requiring frequent data changes