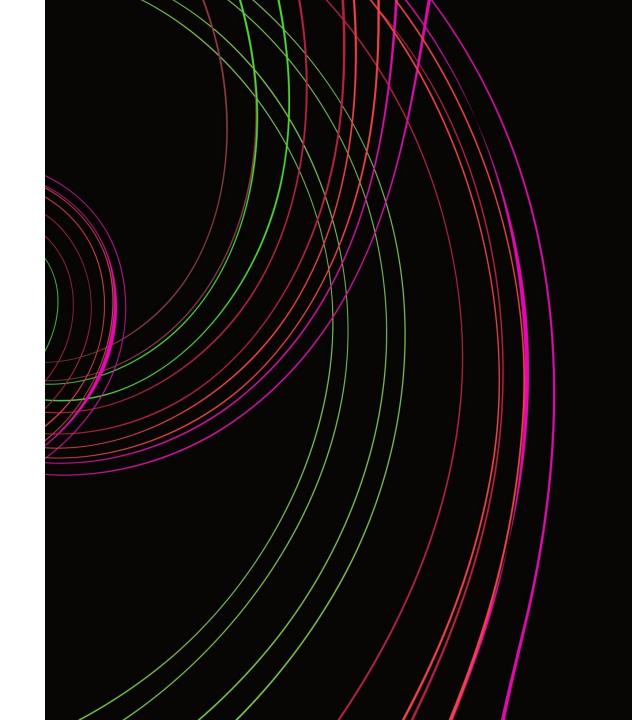
# Embedding Module: From Fundamentals to Deep Representations

DL & GenAl Project [BSDA2001P] Indian Institute of Technology Madras

INDRANIL BHATTACHARYYA

DATA SCIENTIST, RENAULT NISSAN



## Agenda

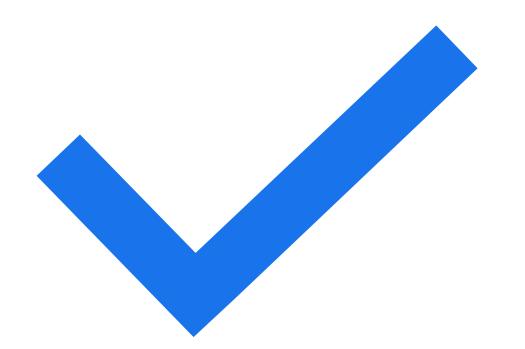
- Setting the Stage: Why GPU for NLP
- From TF-IDF to Deep Embeddings
- Accelerating Embedding Computation with GPUs
- Matryoshka Representation Learning (MRL)
- Wrap-Up & Takeaways

Method	Hardware	Time (per 1k sentences)
TF-IDF	CPU	~2.1s
BERT (base)	CPU	~80s
BERT (base)	GPU	~4.5s

#### Why GPU for NLP?

- Modern NLP models → billions of parameters
   → need parallel tensor operations.
- GPUs accelerate:
  - o Matrix multiplications
  - Batch processing
  - o Embedding generation for large corpora

## Activity – GPU In Kaggle



- Enable GPU in Kaggle
- Verify GPU availability through code

From TF-IDF to Deep Embeddings: The Evolution of Text Representations

#### Traditional Representations: Sparse and Static

$$ext{TF-IDF}(t,d) = ext{tf}(t,d) imes \log rac{N}{df(t)}$$

#### • Conceptual Foundations

- Bag of Words (BoW): Counts term occurrences ignores order & semantics.
- TF-IDF: Weights rare words higher but still independent of context.

#### Limitations:

- High dimensional & sparse vectors
- No notion of semantic similarity
- Fails on polysemous words (e.g., bank → river / finance)

#### Dense Embeddings

- The Transition Phase:
  - o Word2Vec / GloVe: Capture co-occurrence statistics via shallow neural nets.
  - o Learn dense, low-dimensional embeddings (~300D).
  - $\circ$  Each word  $\rightarrow$  a single fixed vector representing global meaning.
- Properties:
  - o Enables vector arithmetic  $\rightarrow$  king man + woman  $\approx$  queen
  - Still static → cannot disambiguate "apple" (fruit vs company)

#### Deep Contextual Representations



Contextual Embeddings with Transformers



**ELMo**, **BERT**, **RoBERTa**: Represent words in *context* using self-attention.



Embedding of a word depends on *surrounding* tokens — **dynamic meaning**.



Multi-layer representations capture hierarchy:

Lower layers → syntax

Middle → semantics

Upper → task-specific nuances

## Activity: Semantic Similarity

- We will use Cosine Similarity to measure the similarity between two sentences.
- Will compare:
  - o Tf-IDF Embedding
  - o Word2Vec
  - o Transformer-based embedding

Deep embeddings compress semantics  $\rightarrow$  fewer dimensions, richer relationships.

Representation	Contextual	Dimensionality	Training	Use-case
BoW/TF-IDF	×	10k+ (Sparse)	None	Simple baselines
Word2Vec	Partial	~300	Self- supervised	Lightweight NLP
BERT / SBERT	<b>~</b>	384–1024	Pre-trained Transformers	Semantic tasks, Sentiment, QA

# Comparative Insights

#### Demo: Visualization Insight

- t-SNE / UMAP:
  - o TF-IDF clusters by *keywords*
  - o BERT clusters by meaning

## Matryoshka Representation Learning (MRL)

## The Problem — Embedding Efficiency at Scale

#### Context:

- o Modern sentence embeddings (e.g., 768–1024D) are **computationally expensive**.
- o Real-world NLP tasks (e.g., retrieval, clustering, QA) don't always need full precision embeddings.
- o Need for compact, multi-resolution embeddings without retraining for every size.

#### Challenge:

o Can we build one embedding space that performs well at multiple dimensionalities?

## What is Matryoshka Representation Learning?

#### **Formal Intuition:**

If  $f(x)\in\mathbb{R}^d$  is the full embedding, then  $f_k(x)=f(x)[:k]$  (the first k dimensions) should maintain meaningful representation quality.

#### Core Idea:

- Like Russian nesting dolls 

   embeddings contain smaller
   embeddings within them.
- A single model is trained so that progressively truncated embeddings (e.g., first 256D of 1024D) still perform well on downstream tasks.
- Key Property: Each prefix of the vector is itself a valid embedding.

## Why this? (Training & Use Case)

- Multi-Scale Training:
  - The model produces a hierarchical embedding vector.
  - During training, multiple truncated versions are supervised to align with the full embedding space.
  - Output embedding: same meaning, smaller footprint.
- Deployment Flexibility:
  - Choose embedding size based on resource constraints:
    - Server  $\rightarrow$  768D
    - Edge device → 256D

## Questions?