**What are tensors?**

**🧱 Step 1: Start simple**

* A **number** (like 5) → a **0-D tensor**.
* A **list of numbers** (like [1,2,3]) → a **1-D tensor** (vector).
* A **table of numbers** (like rows × columns) → a **2-D tensor** (matrix).
* A **stack of tables** → a **3-D tensor** (like an image: height × width × channels).
* Keep stacking → higher-D tensors.

So, a tensor is just a **container for numbers** that can be scalars, vectors, matrices, or multi-dimensional arrays.

**🧮 Step 2: Formal definition**

A **tensor** is a generalization of scalars, vectors, and matrices to any number of dimensions.  
It’s the **main data structure** used in deep learning frameworks (PyTorch, TensorFlow).

**📸 Step 3: Real-life analogy**

* Imagine **LEGO bricks**:
  + A single brick = scalar.
  + A row of bricks = vector.
  + A flat plate (rows × columns) = matrix.
  + A big 3D LEGO cube = 3D tensor.
  + Keep stacking LEGO shapes → higher-D tensors.

**📖 Example in PyTorch**

import torch

# Scalar (0-D tensor)

a = torch.tensor(5)

# Vector (1-D tensor)

b = torch.tensor([1, 2, 3])

# Matrix (2-D tensor)

c = torch.tensor([[1, 2], [3, 4]])

# 3-D tensor (like an RGB image with 2x2 pixels)

d = torch.tensor([[[255, 0, 0], [0, 255, 0]],

[[0, 0, 255], [255, 255, 0]]])

👉 So in your transformer outputs like:

Hidden states shape: torch.Size([1, 12, 768])

that’s a **3-D tensor**:

* 1 → batch size (1 sentence)
* 12 → number of tokens in your input
* 768 → each token’s hidden vector size