What are Tensors

→ A tensor is nothing but a data structure for storing data.
→ If you break down the term, a tensor is primarily a container for numbers.
→ While tensors can sometimes store characters or strings, this is rare (99.99% of the time, they contain numbers).
→ If you have used vectors (lists of numbers) or matrices (2D arrays of numbers), you have already used tensors.
→ Computer scientists, physicists, and mathematicians decided to use the general term "tensor" for structures of N-dimensions.
→ In the computer science community, especially when using NumPy, tensors and N-dimensional arrays (ND-arrays) are considered the same thing.

Key Tensor Properties

→ **Dimension (D):** Refers to how many directions or depths the data has.

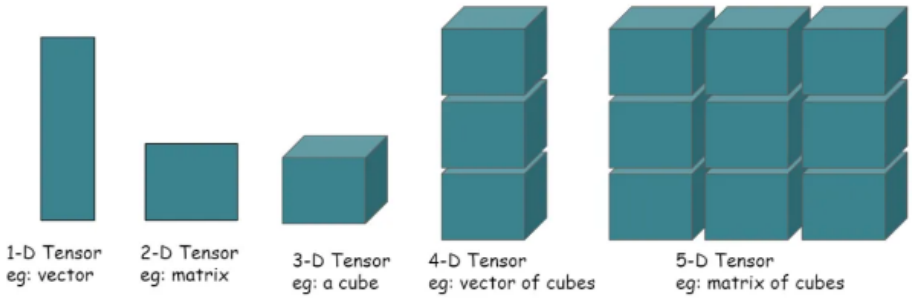
→ **Access/Axes:** Refers to the directions available in the tensor.
- ➔ The number of axes is equal to the number of dimensions.

→ **Rank:** This is a term used interchangeably with the number of dimensions or number of axes of a tensor.

→ **Shape:** Represents the maximum number of items that can be stored along each particular axis.
- ➔ Example: A matrix might have a shape of (2, 3), meaning it can store two items on the row axis and three items on the column axis.

→ **Size:** Represents the total number of items or numbers contained within the tensor.
- ➔ Size is calculated by multiplying all the numbers in the tensor's shape.
- ➔ For a scalar (0D tensor), the size is always 1.

Tensor Hierarchy



Tensors with different dimensions

→ Tensors are built hierarchically: if you connect multiple tensors of a lower dimension, you obtain a tensor of the next higher dimension.
→ A 1D tensor (Vector) is a collection of multiple 0D tensors (scalars).
→ A 2D tensor (Matrix) is a collection of multiple 1D tensors (vectors).
→ A 3D tensor is a collection of 2D tensors (matrices).
→ In practical machine learning and deep learning applications, you generally focus on tensors between 0D and 5D; tensors with higher dimensions are rare.

Practical Tensor Examples (0D to 5D)

1- 0D Tensor (Scalar)
→ A single number is called a scalar or a zero-rank tensor.
→ In NumPy, if a tensor has a dimension of zero (0), it is a 0D tensor.

2- 1D Tensor (Vector)
→ A 1D tensor is a list of numbers, also called an array or a vector.
→ It has a rank of one, meaning it has one dimension/axis.
→ Dimensionality Distinction: While a vector is a 1D tensor, the number of items within the vector defines the dimensionality of the space it occupies (e.g., a vector with three numbers is a 1D tensor but represents a point in a 3D coordinate space).
→ ML Example: The input features for a single data record (e.g., a student's CGPA, IQ, State) are represented as a vector or a 1D tensor. If the input data has 50 features, it remains a 1D tensor but resides in a 50-dimensional coordinate space.

2D Tensor (Matrix)
→ A 2D tensor is a collection of multiple 1D tensors (vectors).
→ It is also known as a Matrix.
→ It has a rank of two, with two axes (a row axis and a column axis).
→ ML Example (Input Data Array): When all the input data for all records (e.g., 10,000 students) is collected together, it forms a Matrix or a 2D Tensor, which is usually called the input data array.

3D Tensor
→ A 3D tensor is a collection of 2D tensors (matrices).
→ It has three axes: column, row, and depth/time.
→ ML Example 1 (NLP/Text): In Natural Language Processing (NLP), after text is converted into numbers (vectorization), a collection of sentences often forms a 3D tensor. If a single sentence is a 2D tensor (a matrix of word vectors), a collection of three sentences forms a 3D tensor.
→ ML Example 2 (Time Series): Data collected over a frequent time period (like daily stock prices) is often a 3D tensor. If a year's worth of data is a 2D tensor, collecting that data over 10 years results in a 3D tensor (10 years × 365 days × quantities), where the middle axis is the time axis.

4D Tensor

→ A 4D tensor is a collection of 3D tensors.

→ ML Example (Batch of Images): This is common in computer vision.

→ A single color image (using Red, Green, and Blue channels) is generally represented as a 3D tensor (e.g., 3 channels × Height × Width).

→ When deep learning models (like Convolutional Neural Networks) process multiple images simultaneously (a batch), the collection of these 3D image tensors forms a 4D Tensor.

5D Tensor

→ A 5D tensor is a collection of 4D tensors.

→ ML Example (Batch of Videos): This is a primary application of 5D tensors.

→ Since a video is essentially a collection of image frames, a single video is already a 4D tensor (Frames × Channels × Height × Width).

→ A collection of multiple videos (e.g., four videos) results in a 5D Tensor.

→ The shape of a 5D video tensor includes: Number of Videos, Number of Images (frames) per video, Channels, Height, and Width.

→ Processing large 5D tensors (like four 60-second color videos) requires significant storage space (potentially over 27 GB uncompressed), which necessitates the use of video encoding formats like MKV or MP4 to reduce data size.

→ Tensors rarely exceed 5D in machine learning or deep learning applications.