Tensor - Data Structures

Deepak H

What is a Tensor? Basic Concepts

At its core, a **tensor** is a multi-dimensional array designed to hold numerical data with a uniform data type.

- Data, in the context of tensors, encompasses anything that can be converted into a numerical form.
- Used for Understanding Complex datas for machine learning
- Tensors can be run on CPUs/GPUs : Enhances Performance
- This image can be represented as a tensor with the shape `[3, 640, 640]`. Here, '3' denotes the number of color channels (Red, Green, Blue), followed by the image's height and width in pixels.

Tensor DataStructure

Rank/ Dimension of a Tensor

- Rank is the Number of indices required to specify each component of a tensor uniquely.
- Defines the number of Dimensions present in a Tensor

```
tens = tf.constant([[[1, 2], [3, 4]], [[5, 6], [7, 8]]])
rank = tf.rank(tens)
```

Shape of a Tensor

- Refers to the layout or structure of a tensor
- defines the number of dimension and the size of each dimension in a tensor.
- Defines how many elements along each axis of a tensor

```
tensor = tf.constant([[1, 2, 3], [4, 5, 6]])
shape = tensor.shape

Shape = (2,3)
```

Type of a Tensor

- Defines the type of values a tenor can hold
- Common data types include int32, float32 and bool
- Determines the precision and memory requirements for storing its element

Tensor Rank

Rank 0 : Scalar

- It represents a single value, such as a real number or a constant.
- Scalars are often used to represent individual values in a mode

Rank 1: Vector

A tensor with a rank of 1 is known as a **vector**. It's a one-dimensional array of numbers.

- Represent a single data point with multiple features

Rank 2: Matrix

A tensor with a rank of 2 is called a **matrix**. This is a two-dimensional array, similar to a spreadsheet.

- represent batch of data, where each row is a sample and each column is a feature

Rank 3: 3D Tensor

A tensor with a rank of 3 is often referred to as a **3D tensor**. You can visualize this as a cube of numbers or a stack of matrices.

- represent data that has spatial or sequential structure

Rank 4: 4D Tensor (Batch Processing)

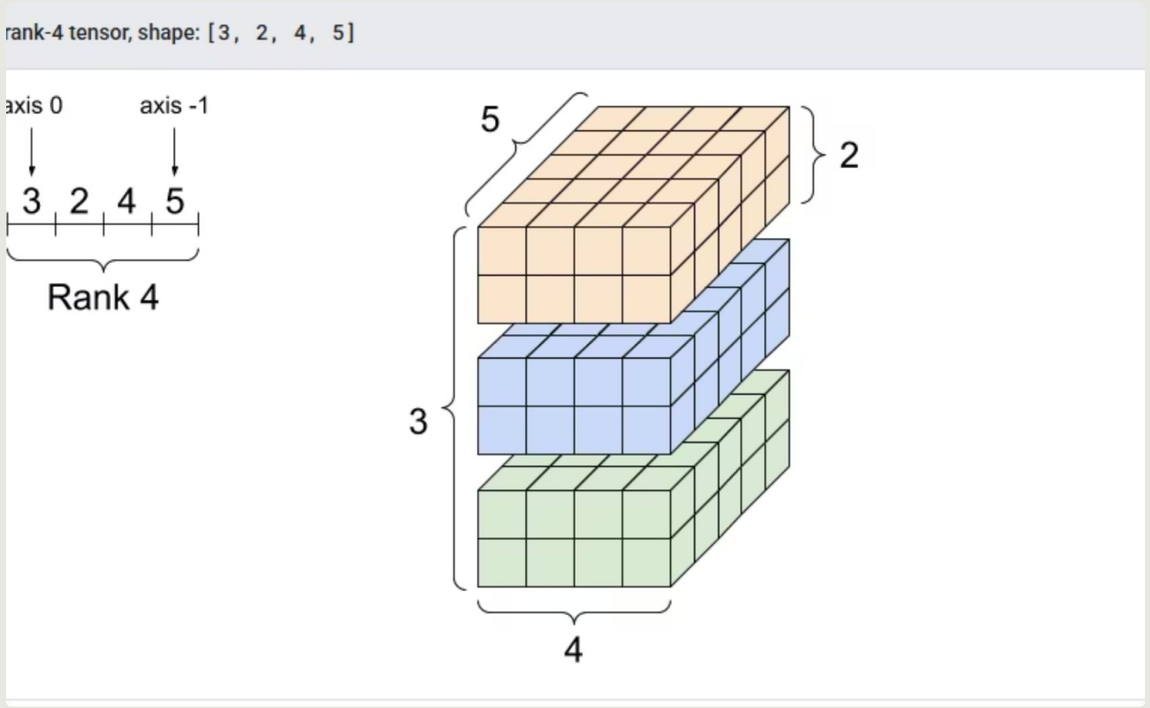
A tensor with a rank of 4 is commonly known as a **4D tensor**. These are frequently used in deep learning, especially when processing images in batches. The typical shape for a batch of images is `[batch_size, height, width, channels]`, where the first dimension represents the number of images in a single processing batch.

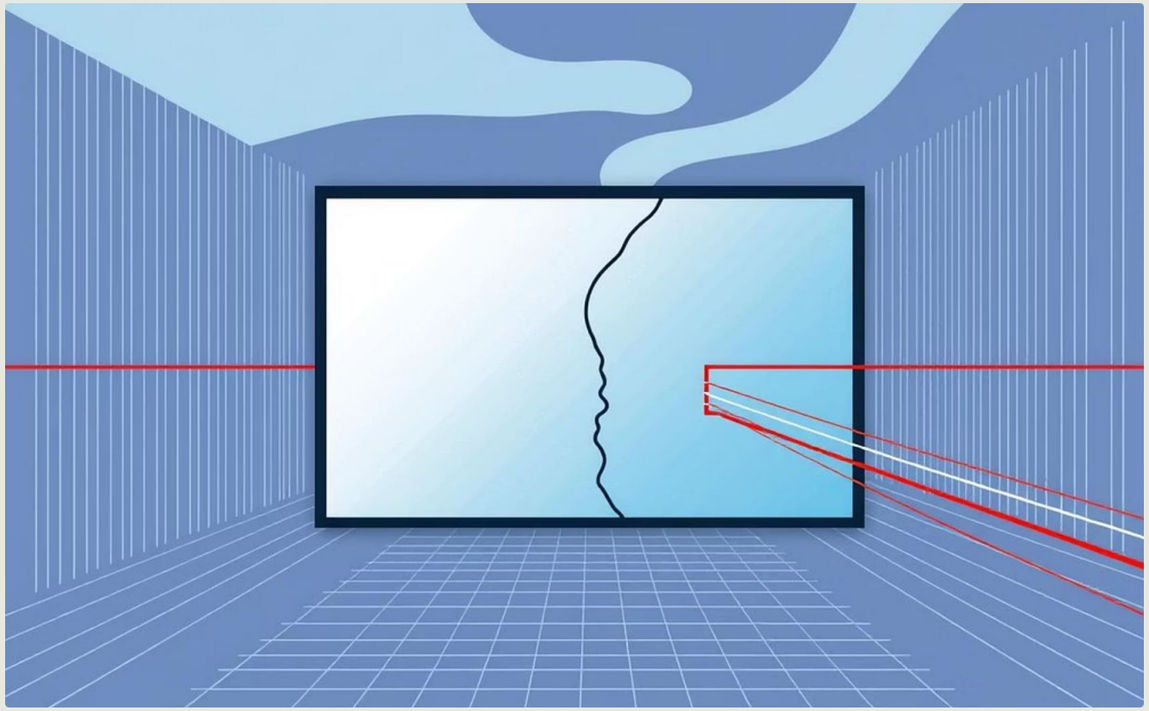
- Tensors can extend to even higher ranks depending on the complexity of the data and the problem at hand.

For example, video data often requires **5D tensors**, where each element might represent a 4D tensor (a single frame) within a sequence of frames, allowing models to process temporal information alongside spatial data.

The Crucial Role of Tensor Shape

Understanding and managing tensor shape is fundamental in machine learning workflows. It defines the structure of your data, and ensuring compatibility is key to successful model building.





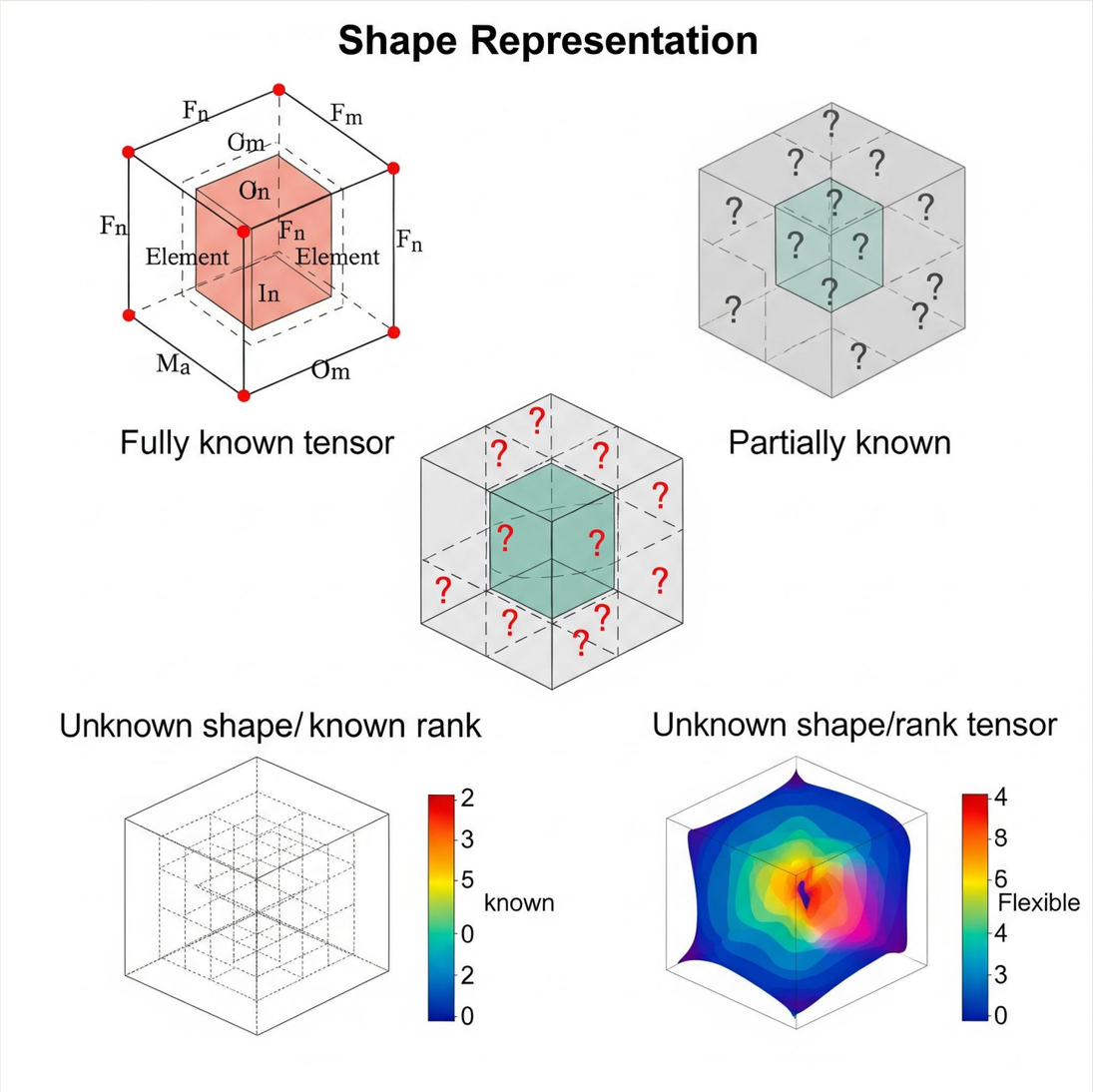
Defining Tensor Shape

The **shape of a tensor** is the length of each axes of a tensor. It specifies its size along each dimension, such as `(2, 3)` for a 2x3 matrix. This structure is critical for how data is processed by algorithms.

The Challenge of Mismatching

A common issue in machine learning is **shape mismatching**. Neural networks are designed to process tensors with exact, predefined shapes. Providing data with incorrect dimensions, like an image of `(64, 64, 3)` to a network expecting `(32, 32, 3)`, will lead to errors.

Tensor’s shape representation



- **Fully-known shape**: Rank and all dimensions are known.
- **Partially-known shape:** Rank known, some dimensions unknown. we just specify the feature vector shape, letting the batch dimension set to *None*
  e.g.: (None, 28, 28, 1).
- **Unknown shape and known rank:** Only the tensor's rank is specified.
  e.g.: (None, None, None)
- **Unknown shape and rank:** Neither rank nor dimensions are known.

Tensor Data Types and Immutability

Common Tensor Data Types

TensorFlow tensors support various data types:

float32: Standard for most neural networks.

int32: For integer labels or indices.

string: For textual data (converted to numerical IDs).

Choosing the right type optimizes memory and computation (e.g., `float16` for speed).

Tensors Are Immutable

TensorFlow tensors are immutable; their values cannot be changed after creation.

Any operation that appears to modify a tensor actually creates a new tensor with the updated values.

This immutability simplifies graph optimization and enables automatic differentiation, vital for neural network backpropagation.

Introducing `tf.Variable` for Mutability

While tensors are immutable, neural networks require mutable state for learnable parameters like **weights** and **biases**. For this, TensorFlow provides `tf.Variable`.

A `tf.Variable` is a special tensor whose value can be modified during training, allowing optimization algorithms to iteratively update model parameters.

Specialized Tensor Structures

TensorFlow also offers specialized tensor types for complex data:

Ragged Tensors: Handle data with non-uniform shapes (e.g., varying sequence lengths).

Sparse Tensors: Efficiently store tensors with many zero values (e.g., in recommendation systems).

Creating and Manipulating Tensors in TensorFlow

TensorFlow provides a comprehensive API for creating and manipulating tensors, forming the bedrock of all numerical computations in machine learning workflows.

Tensor Creation:

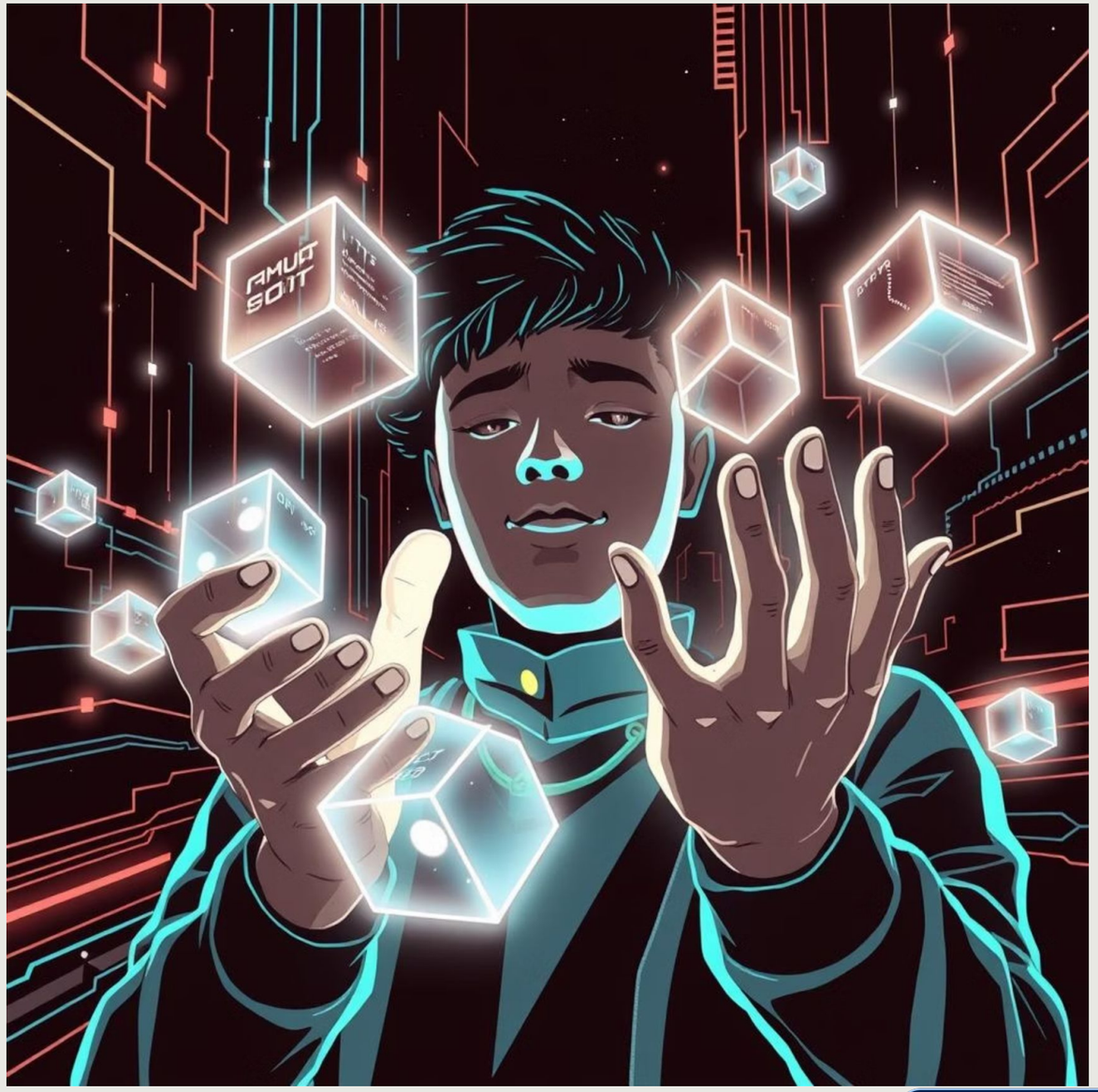
`tf.constant()`: Creates a tensor from a Python list or NumPy array. Ideal for fixed, unchanging data.

`tf.zeros()`: Creates a tensor filled with zeros, useful for initialization or padding.

`tf.ones()`: Creates a tensor filled with ones, also for initialization.

`tf.Variable()`: As discussed, creates a mutable tensor specifically for model parameters that need to be updated.

`tf.random.normal()`, `tf.random.uniform()`: Generate tensors with random values, commonly used for initializing weights in neural networks.



Comparison

Rank	Common Name	Example Shape	Python/TensorFlow Example	Real-World Analogy
0	Scalar	()	tf.constant(5)	A single temperature reading
1	Vector	(3,)	tf.constant()	A list of daily stock prices
2	Matrix	(2, 3)	tf.constant([,])	A grayscale image (height, width)
3	3-Tensor	(10, 224, 224)	tf.zeros((10, 224, 224))	A batch of 10 grayscale images
4	4-Tensor	(10, 224, 224, 3)	tf.zeros((10, 224, 224, 3))	A batch of 10 color images