# Tensors in PyTorch

06 December 2024 12:48

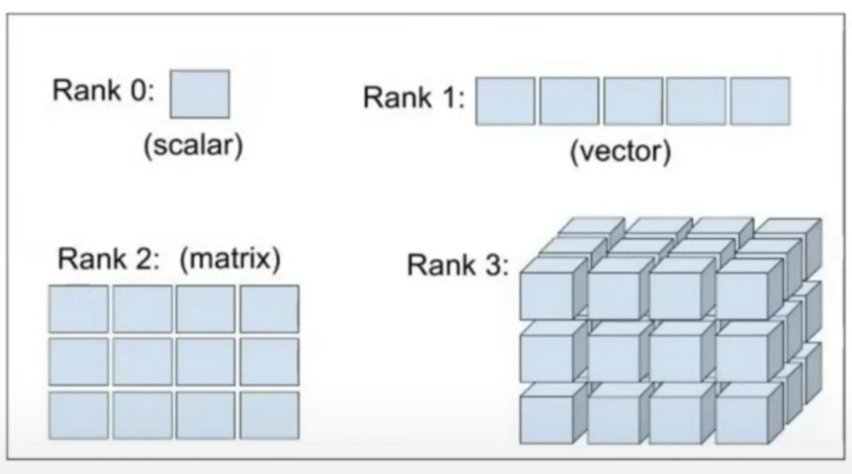
→ Tensors & its Examples.
→ why are tensors useful?
→ where are tensors used in deep learning?

Tensors → are data structures

↳ a way to hold, store & represent data. Ex: arrays. → higher dimensional arrays ↓ tensors.

dimension → direction span of tensor. [ tensor kitne direction mein faila hua hai ]

Tensor is a specialized multi-dimensional array designed for mathematical and computational efficiency.




**Real-World Examples**

1. Scalars: 0-dimensional tensors (a single number)

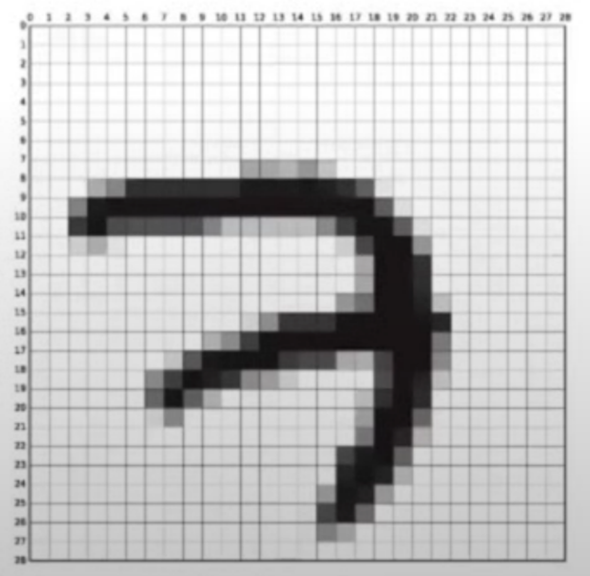
- Represents a single value, often used for simple metrics or constants.
- Example:
  - Loss value: After a forward pass, the loss function computes a single scalar value indicating the difference between the predicted and actual outputs.
  - Example: 5.0 or -3.14 ← loss value [ $\hat{y} - y$ ] ($\hat{y}$) ($y$)

2. Vectors: 1-dimensional tensors (a list of numbers) (arrays)

- Represents a sequence or a collection of values.
- Example:
  - Feature vector: In natural language processing, each word in a sentence may be represented as a 1D vector using embeddings.
  - Example: [0.12, -0.84, 0.33] (a word embedding vector from a pre-trained model like Word2Vec or Glove).

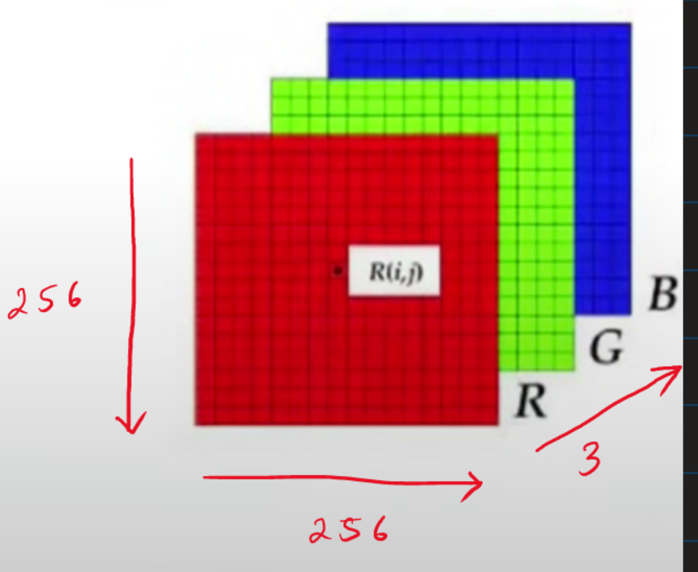
3. Matrices: 2-dimensional tensors (a 2D grid of numbers) ( Grayscale image )

- Represents tabular or grid-like data.
- Example:
  - Grayscale images: A grayscale image can be represented as a 2D tensor, where each entry corresponds to the pixel intensity.
  - Example:
    [[0, 255, 128],
    [34, 90, 180]]



4. 3D Tensors: Coloured images ( RGB ) 3 channel

- Adds a third dimension, often used for stacking data.
- Example:
  - RGB Images: A single RGB image is represented as a 3D tensor (width × height × channels).
  - Examples:
    - RGB Image (e.g., 256x256): Shape [256, 256, 3]




5. 4D Tensors: Batches of RGB images ( Ex: 10 coloured images )

- Adds the batch size as an additional dimension to 3D data.
- Example:
  - Batches of RGB Images: A dataset of coloured images is represented as a 4D tensor (batch size × width × height × channels).
  - Example: A batch of 32 images, each of size 128x128 with 3 colour channels (RGB), would have shape [32, 128, 128, 3].

6. 5D Tensors: Video data [ video data ] [ combination of images/frames → video ]

- Adds a time dimension for data that changes over time (e.g., video frames).
- Example:
  - Video Clips: Represented as a sequence of frames, where each frame is an RGB image.
  - Example: A batch of 10 video clips, each with 16 frames of size 64x64 and 3 channels (RGB), would have shape [10, 16, 64, 64, 3].

**Why Are Tensors Useful?**

1. Mathematical Operations

- Tensors enable efficient mathematical computations (addition, multiplication, dot product, etc.) necessary for neural network operations.

2. Representation of Real-world Data

- Data like images, audio, videos, and text can be represented as tensors:
  - Images: Represented as 3D tensors (width × height × channels).
  - Text: Tokenized and represented as 2D or 3D tensors (sequence length × embedding size).

3. Efficient Computations

- Tensors are optimized for hardware acceleration, allowing computations on GPUs or TPUs, which are crucial for training deep learning models.

↳ Example:

GPU → can be parallelized.

CPU Single Core

Ex: $\begin{bmatrix} 3 & 4 \\ 5 & 6 \end{bmatrix} + \begin{bmatrix} 2 & 5 \\ 7 & 7 \end{bmatrix}$

Single operation at a time.

first these addition will be performed

then this.

Then this

then this

Sequentially.

But since we represented these matrices as tensors we can easily run it on GPUs [ multiple core ]

∴ all the addition can be done parallelly [ handled by diff cores ]

This was a simplified example.

**Where Are Tensors Used in Deep Learning?**

1. Data Storage
   - Training data (images, text, etc.) is stored in tensors.
2. Weights and Biases
   - The learnable parameters of a neural network (weights, biases) are stored as tensors.
3. Matrix Operations
   - Neural networks involve operations like matrix multiplication, dot products, and broadcasting—all performed using tensors.
4. Training Process
   - During forward passes, tensors flow through the network.
   - Gradients, represented as tensors, are calculated during the backward pass.