Tensors

Recap: Linear Algebra

Linear algebra (in deep learning):

A mathematical language to express many operations at once.

Linear Algebra - Examples

Vector

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix}$$

1-dimensional tensor: $\mathbf{v} \in \mathbb{R}^n$

Matrix

$$\mathbf{M} = \begin{bmatrix} M_{1,1} & M_{1,2} & \cdots & M_{1,m} \\ M_{2,1} & M_{2,2} & \cdots & M_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ M_{n,1} & M_{n,2} & \cdots & M_{n,m} \end{bmatrix}$$

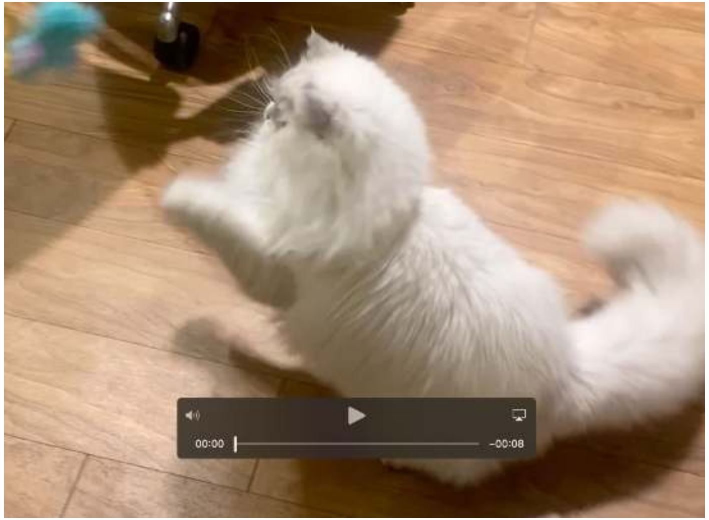
2-dimensional tensor: $\mathbf{M} \in \mathbb{R}^{n \times m}$

What about images / videos?

Image

3-dimensional: $\mathbf{I} \in \mathbb{R}^{H \times W \times C}$

Video



4-dimensional: $\mathbf{V} \in \mathbb{R}^{T \times H \times W \times C}$

Tensors

$$X \in \mathbb{R}^{d_1 \times d_2 \times \cdots \times d_N}$$

X: an N-D array

d_i: the shape along the i-th dimension

Many linear algebra operations can be generalized to tensors

Tensors: Examples

Examples:

- 1D Tensor: Vector
- 2D Tensor: Matrix
- 3D Tensor: Image
- 4D Tensor: Video

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{bmatrix} \qquad \mathbf{M} = \begin{bmatrix} M_{1,1} & M_{1,2} & \cdots & M_{1,m} \\ M_{2,1} & M_{2,2} & \cdots & M_{2,m} \\ \vdots & \vdots & \ddots & \vdots \\ M_{n,1} & M_{n,2} & \cdots & M_{n,m} \end{bmatrix}$$

Tensors in PyTorch

A `torch.Tensor` is a data container.

Important properties:

- `shape` - the size of the tensor
- `dtype` - the data type
- `ndim` - the number of dimensions

Tensors - TL;DR

Tensors are multi-dimensional arrays

Tensors provide an easy and efficient way to represent+manipulate data