

Arrays Vs Tensors

Topic	Array	Tensor
Definition	An array is a data structure consisting of a collection of elements (values or variables), each identified by at least one array index or key.	In the context of data structures, a tensor is a generalization of scalars, vectors and matrices and is easily represented as a multidimensional array.
Dimensions	Arrays can be 1-D (vector), 2-D (matrix), or more.	Tensors extend this concept, can have n-dimensions.
Shape	Shape is determined by the number of elements along each axis, e.g., (3,) for 1D array, (3,2) for 2D array.	Similar to arrays, shape is determined by the number of elements along each axis, can have more axes, e.g., (3,2,2).
Examples	1D array: <code>np.array([1, 2, 3])</code> 2D array: <code>np.array([[1, 2, 3], [4, 5, 6]])</code>	2D tensor (matrix): <code>np.array([[1, 2, 3], [4, 5, 6]])</code> 3D tensor: <code>np.array([[[1, 2, 3], [4, 5, 6]], [[7, 8, 9], [10, 11, 12]]])</code>
Use Cases	Often used in mathematical computations, data manipulation, sorting, etc.	Used in higher dimension computations, often found in Physics, Engineering, Computer Graphics, Machine Learning etc.
Operations	Basic operations include addition, subtraction, multiplication, division, etc.	In addition to basic operations, tensors can also perform high dimensional operations, such as tensor product, tensor contraction, etc.

Note: In programming, especially in Python and TensorFlow, the word "tensor" is often used more generally to mean "n-dimensional array". So, a 1D tensor is a vector, a 2D tensor is a matrix, and anything with 3 or more dimensions is just a tensor.