# Class Tensor

# Table of contents

- Defined in File tensor.hpp

# Class Documentation

class Tensor

Tensor class.

A Tensor is a multi-dimensional array of elements of a single data type.

The Tensor class is a wrapper around the DLManagedTensorContext struct that holds the DLManagedTensor object. (https://dmlc.github.io/dlpack/latest/c_api.html#_CPPv415DLManagedTensor).

This class provides a primary interface to access Tensor data and is interoperable with other frameworks that support DLManagedTensor.

Public Functions

Tensor() = default

inline explicit Tensor(std::shared_ptr<DLManagedTensorContext> &ctx)

Construct a new Tensor from an existing DLManagedTensorContext.

Parameters

**ctx** – A shared pointer to the DLManagedTensorContext to be used in Tensor construction.

explicit Tensor(DLManagedTensor *dl_managed_tensor_ptr)

Construct a new Tensor from an existing DLManagedTensor pointer.

Parameters

**ctx** – A pointer to the DLManagedTensor to be used in Tensor construction.

virtual ~Tensor() = default

inline void *data() const

Get a pointer to the underlying data.

Returns

The pointer to the Tensor's data.

inline DLDevice device() const

Get the device information of the Tensor.

Returns

The device information of the Tensor.

inline DLDataType dtype() const

Get the Tensor's data type information.

For details of the DLDataType struct see the DLPack documentation: https://dmlc.github.io/dlpack/latest/c_api.html#_CPPv410DLDataType

Returns

The DLDataType struct containing DLPack dtype information for the tensor.

std::vector<int64_t> shape() const

Get the shape of the Tensor data.

Returns

The vector containing the Tensor's shape.

std::vector<int64_t> strides() const

Get the strides of the Tensor data.

Note that, unlike `DLTensor.strides`, the strides this method returns are in number of bytes, not elements (to be consistent with NumPy/CuPy's strides).

Returns

The vector containing the Tensor's strides.

bool is_contiguous() const

Check if the tensor a has contiguous, row-major memory layout.

Returns

True if the tensor is contiguous, False otherwise.

int64_t size() const

Get the size (number of elements) in the Tensor.

The size is defined as the number of elements, not the number of bytes. For the latter, see nbytes.

If the underlying DLDataType contains multiple lanes, all lanes are considered as a single element. For example, a float4 vectorized type is counted as a single element, not four elements.

Returns

The size of the tensor in number of elements.

inline int32_t ndim() const

Get the number of dimensions of the Tensor.

Returns

The number of dimensions.

inline uint8_t itemsize() const

Get the itemsize of a single Tensor data element.

If the underlying DLDataType contains multiple lanes, itemsize takes this into account. For example, a Tensor containing (vectorized) float4 elements would have itemsize 16, not 4.

Returns

The itemsize of the Tensor's data.

inline int64_t nbytes() const

Get the total number of bytes for the Tensor's data.

Returns

The size of the Tensor's data in bytes.

DLManagedTensor *to_dlpack()

Get a DLPack managed tensor pointer to the Tensor.

Returns

A DLManagedTensor* pointer corresponding to the Tensor.

inline std::shared_ptr<DLManagedTensorContext> &dl_ctx()

Get the internal DLManagedTensorContext of the Tensor.

Returns

A shared pointer to the Tensor's DLManagedTensorContext.

Protected Attributes

std::shared_ptr<DLManagedTensorContext> dl_ctx_

The DLManagedTensorContext object.