torch.Tensor

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\(pytorch-master)(docs)(source)tensors.rst`, **line 1**)
>
> Unknown directive type "currentmodule".
>
> ```
> .. currentmodule:: torch
> ```

A :class:`torch.Tensor` is a multi-dimensional matrix containing elements of a single data type.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\(pytorch-master)(docs)(source)tensors.rst`, **line 8**); *backlink*
>
> Unknown interpreted text role "class".

Data types

Torch defines 10 tensor types with CPU and GPU variants which are as follows:

Data type	dtype	CPU tensor	GPU tensor
32-bit floating point	`torch.float32` or `torch.float`	:class:`torch.FloatTensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\(pytorch-master)(docs)(source)tensors.rst`, **line 21**); *backlink* Unknown interpreted text role "class".	:class:`torch.cuda.FloatTensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\(pytorch-master)(docs)(source)tensors.rst`, **line 21**); *backlink* Unknown interpreted text role "class".
64-bit floating point	`torch.float64` or `torch.double`	:class:`torch.DoubleTensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\(pytorch-master)(docs)(source)tensors.rst`, **line 22**); *backlink* Unknown interpreted text role "class".	:class:`torch.cuda.DoubleTensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\(pytorch-master)(docs)(source)tensors.rst`, **line 22**); *backlink* Unknown interpreted text role "class".

Data type	dtype	CPU tensor	GPU tensor
16-bit floating point [1]	`torch.float16` or `torch.half`	:class:`torch.HalfTensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\` `(pytorch-master)` `(docs)` `(source)tensors.rst,` **line 23);** *backlink* Unknown interpreted text role "class".	:class:`torch.cuda.HalfTensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\` `(pytorch-master)` `(docs)` `(source)tensors.rst,` **line 23);** *backlink* Unknown interpreted text role "class".
16-bit floating point [2]	`torch.bfloat16`	:class:`torch.BFloat16Tensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\` `(pytorch-master)` `(docs)` `(source)tensors.rst,` **line 24);** *backlink* Unknown interpreted text role "class".	:class:`torch.cuda.BFloat16Tensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\` `(pytorch-master)` `(docs)` `(source)tensors.rst,` **line 24);** *backlink* Unknown interpreted text role "class".
32-bit complex	`torch.complex32`		
64-bit complex	`torch.complex64`		
128-bit complex	`torch.complex128` or `torch.cdouble`		
8-bit integer (unsigned)	`torch.uint8`	:class:`torch.ByteTensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\` `(pytorch-master)` `(docs)` `(source)tensors.rst,` **line 28);** *backlink* Unknown interpreted text role "class".	:class:`torch.cuda.ByteTensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\` `(pytorch-master)` `(docs)` `(source)tensors.rst,` **line 28);** *backlink* Unknown interpreted text role "class".

Data type	dtype	CPU tensor	GPU tensor
8-bit integer (signed)	`torch.int8`	:class:`torch.CharTensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\` `(pytorch-master)` `(docs)` `(source)tensors.rst,` **line 29);** *backlink* Unknown interpreted text role "class".	:class:`torch.cuda.CharTensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\` `(pytorch-master)` `(docs)` `(source)tensors.rst,` **line 29);** *backlink* Unknown interpreted text role "class".
16-bit integer (signed)	`torch.int16` or `torch.short`	:class:`torch.ShortTensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\` `(pytorch-master)` `(docs)` `(source)tensors.rst,` **line 30);** *backlink* Unknown interpreted text role "class".	:class:`torch.cuda.ShortTensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\` `(pytorch-master)` `(docs)` `(source)tensors.rst,` **line 30);** *backlink* Unknown interpreted text role "class".
32-bit integer (signed)	`torch.int32` or `torch.int`	:class:`torch.IntTensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\` `(pytorch-master)` `(docs)` `(source)tensors.rst,` **line 31);** *backlink* Unknown interpreted text role "class".	:class:`torch.cuda.IntTensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\` `(pytorch-master)` `(docs)` `(source)tensors.rst,` **line 31);** *backlink* Unknown interpreted text role "class".
		:class:`torch.CharTensor`	:class:`torch.cuda.CharTensor`

Data type	dtype	CPU tensor	GPU tensor
64-bit integer (signed)	`torch.int64` or `torch.long`	:class:`torch.LongTensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\` `(pytorch-master)` `(docs)` `(source)tensors.rst,` **line 32);** *backlink* Unknown interpreted text role "class".	:class:`torch.cuda.LongTensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\` `(pytorch-master)` `(docs)` `(source)tensors.rst,` **line 32);** *backlink* Unknown interpreted text role "class".
Boolean	`torch.bool`	:class:`torch.BoolTensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\` `(pytorch-master)` `(docs)` `(source)tensors.rst,` **line 33);** *backlink* Unknown interpreted text role "class".	:class:`torch.cuda.BoolTensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\` `(pytorch-master)` `(docs)` `(source)tensors.rst,` **line 33);** *backlink* Unknown interpreted text role "class".
quantized 8-bit integer (unsigned)	`torch.quint8`	:class:`torch.ByteTensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\` `(pytorch-master)` `(docs)` `(source)tensors.rst,` **line 34);** *backlink* Unknown interpreted text role "class".	/

Data type	dtype	CPU tensor	GPU tensor
quantized 8-bit integer (signed)	torch.qint8	:class:`torch.CharTensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\ (pytorch-master) (docs) (source) tensors.rst,` **line 35);** *backlink* Unknown interpreted text role "class".	/
quantized 32-bit integer (signed)	torch.qint32	:class:`torch.IntTensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\ (pytorch-master) (docs) (source) tensors.rst,` **line 36);** *backlink* Unknown interpreted text role "class".	/
quantized 4-bit integer (unsigned) [3]	torch.quint4x2	:class:`torch.ByteTensor` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\ (pytorch-master) (docs) (source) tensors.rst,` **line 37);** *backlink* Unknown interpreted text role "class".	/

[1] Sometimes referred to as binary16: uses 1 sign, 5 exponent, and 10 significand bits. Useful when precision is important at the expense of range.

[2] Sometimes referred to as Brain Floating Point: uses 1 sign, 8 exponent, and 7 significand bits. Useful when range is important, since it has the same number of exponent bits as `float32`

[3] quantized 4-bit integer is stored as a 8-bit signed integer. Currently it's only supported in EmbeddingBag operator.

:class:`torch.Tensor` is an alias for the default tensor type (:class:`torch.FloatTensor`).

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\ (pytorch-master) (docs) (source) tensors.rst,` **line 49);** *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\(pytorch-master)(docs)(source)tensors.rst`, line 49); *backlink*

Unknown interpreted text role "class".

Initializing and basic operations

A tensor can be constructed from a Python :class:`list` or sequence using the :func:`torch.tensor` constructor:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\(pytorch-master)(docs)(source)tensors.rst`, line 54); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\(pytorch-master)(docs)(source)tensors.rst`, line 54); *backlink*

Unknown interpreted text role "func".

```
>>> torch.tensor([[1., -1.], [1., -1.]])
tensor([[ 1.0000, -1.0000],
        [ 1.0000, -1.0000]])
>>> torch.tensor(np.array([[1, 2, 3], [4, 5, 6]]))
tensor([[ 1,  2,  3],
        [ 4,  5,  6]])
```

Warning

:func:`torch.tensor` always copies :attr:`data`. If you have a Tensor :attr:`data` and just want to change its `requires_grad` flag, use :meth:`~torch.Tensor.requires_grad_` or :meth:`~torch.Tensor.detach` to avoid a copy. If you have a numpy array and want to avoid a copy, use :func:`torch.as_tensor`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\(pytorch-master)(docs)(source)tensors.rst`, line 68); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\(pytorch-master)(docs)(source)tensors.rst`, line 68); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\(pytorch-master)(docs)(source)tensors.rst`, line 68); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\(pytorch-master)(docs)(source)tensors.rst`, line 68); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\(pytorch-master)(docs)(source)tensors.rst`, line 68); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\(pytorch-master)(docs)(source)tensors.rst`, line 68); *backlink*

Unknown interpreted text role "func".

A tensor of specific data type can be constructed by passing a :class:`torch.dtype` and/or a :class:`torch.device` to a constructor or tensor creation op:

```
>>> torch.zeros([2, 4], dtype=torch.int32)
tensor([[ 0,  0,  0,  0],
        [ 0,  0,  0,  0]], dtype=torch.int32)
>>> cuda0 = torch.device('cuda:0')
>>> torch.ones([2, 4], dtype=torch.float64, device=cuda0)
tensor([[ 1.0000,  1.0000,  1.0000,  1.0000],
        [ 1.0000,  1.0000,  1.0000,  1.0000]], dtype=torch.float64, device='cuda:0')
```

For more information about building Tensors, see :ref:`tensor-creation-ops`

The contents of a tensor can be accessed and modified using Python's indexing and slicing notation:

```
>>> x = torch.tensor([[1, 2, 3], [4, 5, 6]])
>>> print(x[1][2])
tensor(6)
>>> x[0][1] = 8
>>> print(x)
tensor([[ 1,  8,  3],
        [ 4,  5,  6]])
```

Use :meth:`torch.Tensor.item` to get a Python number from a tensor containing a single value:

```
>>> x = torch.tensor([[1]])
>>> x
tensor([[ 1]])
>>> x.item()
1
>>> x = torch.tensor(2.5)
>>> x
tensor(2.5000)
>>> x.item()
2.5
```

For more information about indexing, see :ref:`indexing-slicing-joining`

A tensor can be created with :attr:`requires_grad=True` so that :mod:`torch.autograd` records operations on them for automatic differentiation.

Unknown interpreted text role "attr".

```
>>> x = torch.tensor([[1., -1.], [1., 1.]], requires_grad=True)
>>> out = x.pow(2).sum()
>>> out.backward()
>>> x.grad
tensor([[ 2.0000, -2.0000],
        [ 2.0000,  2.0000]])
```

Each tensor has an associated :class:`torch.Storage`, which holds its data. The tensor class also provides multi-dimensional, strided view of a storage and defines numeric operations on it.

Note

For more information on tensor views, see :ref:`tensor-view-doc`.

Note

For more information on the :class:`torch.dtype`, :class:`torch.device`, and :class:`torch.layout` attributes of a :class:`torch.Tensor`, see :ref:`tensor-attributes-doc`.

Unknown interpreted text role "ref".

Note

Methods which mutate a tensor are marked with an underscore suffix. For example, :func:`torch.FloatTensor.abs_` computes the absolute value in-place and returns the modified tensor, while :func:`torch.FloatTensor.abs` computes the result in a new tensor.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\(pytorch-master)(docs) (source)tensors.rst`, **line 148**); *backlink*
>
> Unknown interpreted text role "func".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\(pytorch-master)(docs) (source)tensors.rst`, **line 148**); *backlink*
>
> Unknown interpreted text role "func".

Note

To change an existing tensor's :class:`torch.device` and/or :class:`torch.dtype`, consider using :meth:`~torch.Tensor.to` method on the tensor.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\(pytorch-master)(docs) (source)tensors.rst`, **line 154**); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\(pytorch-master)(docs) (source)tensors.rst`, **line 154**); *backlink*
>
> Unknown interpreted text role "class".

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\(pytorch-master)(docs) (source)tensors.rst`, **line 154**); *backlink*
>
> Unknown interpreted text role "meth".

Warning

Current implementation of :class:`torch.Tensor` introduces memory overhead, thus it might lead to unexpectedly high memory usage in the applications with many tiny tensors. If this is your case, consider using one large structure.

> **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\(pytorch-master)(docs) (source)tensors.rst`, **line 158**); *backlink*
>
> Unknown interpreted text role "class".

Tensor class reference

There are a few main ways to create a tensor, depending on your use case.

- To create a tensor with pre-existing data, use :func:`torch.tensor`.

- To create a tensor with specific size, use `torch.*` tensor creation ops (see :ref:`tensor-creation-ops`).

- To create a tensor with the same size (and similar types) as another tensor, use `torch.*_like` tensor creation ops (see :ref:`tensor-creation-ops`).

- To create a tensor with similar type but different size as another tensor, use `tensor.new_*` creation ops.

```
Tensor.is_cuda
Tensor.is_quantized
Tensor.is_meta
Tensor.device
Tensor.grad
Tensor.ndim
Tensor.real
Tensor.imag

Tensor.abs
Tensor.abs_
Tensor.absolute
Tensor.absolute_
Tensor.acos
Tensor.acos_
Tensor.arccos
Tensor.arccos_
Tensor.add
Tensor.add_
Tensor.addbmm
Tensor.addbmm_
Tensor.addcdiv
Tensor.addcdiv_
Tensor.addcmul
Tensor.addcmul_
Tensor.addmm
Tensor.addmm_
Tensor.sspaddmm
Tensor.addmv
Tensor.addmv_
Tensor.addr
Tensor.addr_
Tensor.adjoint
Tensor.allclose
Tensor.amax
Tensor.amin
Tensor.aminmax
Tensor.angle
Tensor.apply_
Tensor.argmax
Tensor.argmin
Tensor.argsort
Tensor.argwhere
Tensor.asin
Tensor.asin_
Tensor.arcsin
Tensor.arcsin_
Tensor.as_strided
Tensor.atan
Tensor.atan_
Tensor.arctan
Tensor.arctan_
Tensor.atan2
Tensor.atan2_
Tensor.arctan2
Tensor.arctan2_
Tensor.all
Tensor.any
Tensor.backward
Tensor.baddbmm
Tensor.baddbmm_
Tensor.bernoulli
Tensor.bernoulli_
Tensor.bfloat16
Tensor.bincount
Tensor.bitwise_not
Tensor.bitwise_not_
Tensor.bitwise_and
Tensor.bitwise_and_
Tensor.bitwise_or
Tensor.bitwise_or_
Tensor.bitwise_xor
Tensor.bitwise_xor_
Tensor.bitwise_left_shift
Tensor.bitwise_left_shift_
Tensor.bitwise_right_shift
Tensor.bitwise_right_shift_
Tensor.bmm
Tensor.bool
Tensor.byte
```
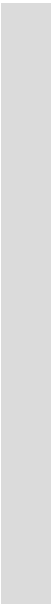
```
Tensor.broadcast_to
Tensor.cauchy_
Tensor.ceil
Tensor.ceil_
Tensor.char
Tensor.cholesky
Tensor.cholesky_inverse
Tensor.cholesky_solve
Tensor.chunk
Tensor.clamp
Tensor.clamp_
Tensor.clip
Tensor.clip_
Tensor.clone
Tensor.contiguous
Tensor.copy_
Tensor.conj
Tensor.conj_physical
Tensor.conj_physical_
Tensor.resolve_conj
Tensor.resolve_neg
Tensor.copysign
Tensor.copysign_
Tensor.cos
Tensor.cos_
Tensor.cosh
Tensor.cosh_
Tensor.corrcoef
Tensor.count_nonzero
Tensor.cov
Tensor.acosh
Tensor.acosh_
Tensor.arccosh
Tensor.arccosh_
Tensor.cpu
Tensor.cross
Tensor.cuda
Tensor.logcumsumexp
Tensor.cummax
Tensor.cummin
Tensor.cumprod
Tensor.cumprod_
Tensor.cumsum
Tensor.cumsum_
Tensor.data_ptr
Tensor.deg2rad
Tensor.dequantize
Tensor.det
Tensor.dense_dim
Tensor.detach
Tensor.detach_
Tensor.diag
Tensor.diag_embed
Tensor.diagflat
Tensor.diagonal
Tensor.diagonal_scatter
Tensor.fill_diagonal_
Tensor.fmax
Tensor.fmin
Tensor.diff
Tensor.digamma
Tensor.digamma_
Tensor.dim
Tensor.dist
Tensor.div
Tensor.div_
Tensor.divide
Tensor.divide_
Tensor.dot
Tensor.double
Tensor.dsplit
Tensor.eig
Tensor.element_size
Tensor.eq
Tensor.eq_
Tensor.equal
Tensor.erf
Tensor.erf_
Tensor.erfc
Tensor.erfc_
Tensor.erfinv
```

```
Tensor.erfinv_
Tensor.exp
Tensor.exp_
Tensor.expm1
Tensor.expm1_
Tensor.expand
Tensor.expand_as
Tensor.exponential_
Tensor.fix
Tensor.fix_
Tensor.fill_
Tensor.flatten
Tensor.flip
Tensor.fliplr
Tensor.flipud
Tensor.float
Tensor.float_power
Tensor.float_power_
Tensor.floor
Tensor.floor_
Tensor.floor_divide
Tensor.floor_divide_
Tensor.fmod
Tensor.fmod_
Tensor.frac
Tensor.frac_
Tensor.frexp
Tensor.gather
Tensor.gcd
Tensor.gcd_
Tensor.ge
Tensor.ge_
Tensor.greater_equal
Tensor.greater_equal_
Tensor.geometric_
Tensor.geqrf
Tensor.ger
Tensor.get_device
Tensor.gt
Tensor.gt_
Tensor.greater
Tensor.greater_
Tensor.half
Tensor.hardshrink
Tensor.heaviside
Tensor.histc
Tensor.histogram
Tensor.hsplit
Tensor.hypot
Tensor.hypot_
Tensor.i0
Tensor.i0_
Tensor.igamma
Tensor.igamma_
Tensor.igammac
Tensor.igammac_
Tensor.index_add_
Tensor.index_add
Tensor.index_copy_
Tensor.index_copy
Tensor.index_fill_
Tensor.index_fill
Tensor.index_put_
Tensor.index_put
Tensor.index_select
Tensor.indices
Tensor.inner
Tensor.int
Tensor.int_repr
Tensor.inverse
Tensor.isclose
Tensor.isfinite
Tensor.isinf
Tensor.isposinf
Tensor.isneginf
Tensor.isnan
Tensor.is_contiguous
Tensor.is_complex
Tensor.is_conj
Tensor.is_floating_point
Tensor.is_inference
```

```
Tensor.is_leaf
Tensor.is_pinned
Tensor.is_set_to
Tensor.is_shared
Tensor.is_signed
Tensor.is_sparse
Tensor.istft
Tensor.isreal
Tensor.item
Tensor.kthvalue
Tensor.lcm
Tensor.lcm_
Tensor.ldexp
Tensor.ldexp_
Tensor.le
Tensor.le_
Tensor.less_equal
Tensor.less_equal_
Tensor.lerp
Tensor.lerp_
Tensor.lgamma
Tensor.lgamma_
Tensor.log
Tensor.log_
Tensor.logdet
Tensor.log10
Tensor.log10_
Tensor.log1p
Tensor.log1p_
Tensor.log2
Tensor.log2_
Tensor.log_normal_
Tensor.logaddexp
Tensor.logaddexp2
Tensor.logsumexp
Tensor.logical_and
Tensor.logical_and_
Tensor.logical_not
Tensor.logical_not_
Tensor.logical_or
Tensor.logical_or_
Tensor.logical_xor
Tensor.logical_xor_
Tensor.logit
Tensor.logit_
Tensor.long
Tensor.lstsq
Tensor.lt
Tensor.lt_
Tensor.less
Tensor.less_
Tensor.lu
Tensor.lu_solve
Tensor.as_subclass
Tensor.map_
Tensor.masked_scatter_
Tensor.masked_scatter
Tensor.masked_fill_
Tensor.masked_fill
Tensor.masked_select
Tensor.matmul
Tensor.matrix_power
Tensor.matrix_exp
Tensor.max
Tensor.maximum
Tensor.mean
Tensor.nanmean
Tensor.median
Tensor.nanmedian
Tensor.min
Tensor.minimum
Tensor.mm
Tensor.smm
Tensor.mode
Tensor.movedim
Tensor.moveaxis
Tensor.msort
Tensor.mul
Tensor.mul_
Tensor.multiply
Tensor.multiply_
```

```
Tensor.multinomial
Tensor.mv
Tensor.mvlgamma
Tensor.mvlgamma_
Tensor.nansum
Tensor.narrow
Tensor.narrow_copy
Tensor.ndimension
Tensor.nan_to_num
Tensor.nan_to_num_
Tensor.ne
Tensor.ne_
Tensor.not_equal
Tensor.not_equal_
Tensor.neg
Tensor.neg_
Tensor.negative
Tensor.negative_
Tensor.nelement
Tensor.nextafter
Tensor.nextafter_
Tensor.nonzero
Tensor.norm
Tensor.normal_
Tensor.numel
Tensor.numpy
Tensor.orgqr
Tensor.ormqr
Tensor.outer
Tensor.permute
Tensor.pin_memory
Tensor.pinverse
Tensor.polygamma
Tensor.polygamma_
Tensor.positive
Tensor.pow
Tensor.pow_
Tensor.prod
Tensor.put_
Tensor.qr
Tensor.qscheme
Tensor.quantile
Tensor.nanquantile
Tensor.q_scale
Tensor.q_zero_point
Tensor.q_per_channel_scales
Tensor.q_per_channel_zero_points
Tensor.q_per_channel_axis
Tensor.rad2deg
Tensor.random_
Tensor.ravel
Tensor.reciprocal
Tensor.reciprocal_
Tensor.record_stream
Tensor.register_hook
Tensor.remainder
Tensor.remainder_
Tensor.renorm
Tensor.renorm_
Tensor.repeat
Tensor.repeat_interleave
Tensor.requires_grad
Tensor.requires_grad_
Tensor.reshape
Tensor.reshape_as
Tensor.resize_
Tensor.resize_as_
Tensor.retain_grad
Tensor.retains_grad
Tensor.roll
Tensor.rot90
Tensor.round
Tensor.round_
Tensor.rsqrt
Tensor.rsqrt_
Tensor.scatter
Tensor.scatter_
Tensor.scatter_add_
Tensor.scatter_add
Tensor.scatter_reduce_
Tensor.scatter_reduce
```

```
Tensor.select
Tensor.select_scatter
Tensor.set_
Tensor.share_memory_
Tensor.short
Tensor.sigmoid
Tensor.sigmoid_
Tensor.sign
Tensor.sign_
Tensor.signbit
Tensor.sgn
Tensor.sgn_
Tensor.sin
Tensor.sin_
Tensor.sinc
Tensor.sinc_
Tensor.sinh
Tensor.sinh_
Tensor.asinh
Tensor.asinh_
Tensor.arcsinh
Tensor.arcsinh_
Tensor.size
Tensor.slogdet
Tensor.slice_scatter
Tensor.solve
Tensor.sort
Tensor.split
Tensor.sparse_mask
Tensor.sparse_dim
Tensor.sqrt
Tensor.sqrt_
Tensor.square
Tensor.square_
Tensor.squeeze
Tensor.squeeze_
Tensor.std
Tensor.stft
Tensor.storage
Tensor.storage_offset
Tensor.storage_type
Tensor.stride
Tensor.sub
Tensor.sub_
Tensor.subtract
Tensor.subtract_
Tensor.sum
Tensor.sum_to_size
Tensor.svd
Tensor.swapaxes
Tensor.swapdims
Tensor.symeig
Tensor.t
Tensor.t_
Tensor.tensor_split
Tensor.tile
Tensor.to
Tensor.to_mkldnn
Tensor.take
Tensor.take_along_dim
Tensor.tan
Tensor.tan_
Tensor.tanh
Tensor.tanh_
Tensor.atanh
Tensor.atanh_
Tensor.arctanh
Tensor.arctanh_
Tensor.tolist
Tensor.topk
Tensor.to_sparse
Tensor.trace
Tensor.transpose
Tensor.transpose_
Tensor.triangular_solve
Tensor.tril
Tensor.tril_
Tensor.triu
Tensor.triu_
Tensor.true_divide
Tensor.true_divide_
```

```
Tensor.trunc
Tensor.trunc_
Tensor.type
Tensor.type_as
Tensor.unbind
Tensor.unfold
Tensor.uniform_
Tensor.unique
Tensor.unique_consecutive
Tensor.unsqueeze
Tensor.unsqueeze_
Tensor.values
Tensor.var
Tensor.vdot
Tensor.view
Tensor.view_as
Tensor.vsplit
Tensor.where
Tensor.xlogy
Tensor.xlogy_
Tensor.zero_
```