Ta le of Contents

torch

The torch package contains at a structures for multi- imensional tensors an efines mathematical operations over these tensors.

A itionally, it provi es many utilities for efficient serialization of Tensors an ar itrary types, an other useful utilities.

It has a C DA counterpart, that enalles you to run your tensor computations on an NVIDIA GP with compute capalility >= 3.0.

Tensors

is_tensor	Returns True if <i>obj</i> is a PyTorch tensor.
is_storage	Returns True if <i>obj</i> is a PyTorch storage o ect.
is_complex	Returns True if the ata type of input is a complex ata type i.e., one of torch.complex64, an torch.complex128.
is_conj	Returns True if the input is a con ugate tensor, i.e. its con ugate it is set to <i>True</i> .
is_floating_point	Returns True if the ata type of input is a floating point ata type i.e., one of torch.float64, torch.float32, torch.float16, an torch.bfloat16.
is_nonzero	Returns True if the input is a single element tensor which is not equal to zero after type conversions.
set_default_dtype	Sets the efault floating point type to d.
get_default_dtype	Get the current efault floating point torch.dtype.
set_default_device	Sets the efault torch.Tensor to e allocate on device.
get_default_device	Gets the efault torch.Tensor to e allocate on device
set_default_tensor_type	
numel	Returns the total num er of elements in the input tensor.
set_printoptions	Set options for printing.
set_flush_denormal	Disa les enormal floating num ers on CP .

Creation O s

• NOTE	
<pre>torch.randn_like() torch.randint() torch.rand</pre>	<pre>sampling an inclu e: torch.rand() torch.rand_like() torch.randn() dint_like() torch.randperm() You may also use torch.empty() with the nsor s with values sample from a roa er range of istri utions.</pre>
tensor	Constructs a tensor with no autogra history (also known as a "leaf tensor", see Autogra mechanics) y copying data.
sparse_coo_tensor	Constructs a sparse tensor in COO(r inate) format with specifie values at the given indices.
sparse_csr_tensor	Constructs a sparse tensor in CSR (Compresse Sparse Row) with specifie values at the given crow_indices an col_indices.
sparse_csc_tensor	Constructs a sparse tensor in CSC (Compresse Sparse Column) with specifie values at the given ccol_indices an row_indices.
sparse_bsr_tensor	Constructs a sparse tensor in BSR (Block Compresse Sparse Row)) with specifie 2- imensional locks at the given crow_indices an col_indices.
sparse_bsc_tensor	Constructs a sparse tensor in BSC (Block Compresse Sparse Column)) with specifie 2- imensional locks at the given ccol_indices an row_indices.
asarray	Converts obj to a tensor.

as_strided	size, stride an storage_offset.
from_file	Creates a CP tensor with a storage acke y a memory-mappe file.
from_numpy	Creates a Tensor from a numpy.ndarray.
from_dlpack	Converts a tensor from an external li rary into a torch. Tensor.
frombuffer	Creates a 1- imensional Tensor from an o ect that implements the Python uffer protocol.
zeros	Returns a tensor fille with the scalar value o, with the shape efine y the varia le argument size.
zeros_like	Returns a tensor fille with the scalar value o, with the same size as input.
ones	Returns a tensor fille with the scalar value 1, with the shape efine y the varia le argument size.
ones_like	Returns a tensor fille with the scalar value 1, with the same size as input.
arange	Returns a 1-D tensor of size $\left\lceil \frac{\text{end-start}}{\text{step}} \right\rceil$ with values from the interval [start, end) taken with common ifference step eginning from start.
range	Returns a 1-D tensor of size $\left\lfloor \frac{\mathrm{end-start}}{\mathrm{step}} \right\rfloor + 1$ with values from start to end with step step.
linspace	Creates a one- imensional tensor of size steps whose values are evenly space from start to end, inclusive.
logspace	Creates a one- imensional tensor of size steps whose values are evenly space from $base^{start}$ to $base^{end}$, inclusive, on a logarithmic scale with $\ \ ase\ \ base$.
eye	Returns a 2-D tensor with ones on the iagonal an zeros elsewhere.
empty	Returns a tensor fille with uninitialize ata.
empty_like	Returns an uninitialize tensor with the same size as input.
empty_strided	Creates a tensor with the specifie size an stride an fille with un efine ata.
full	Creates a tensor of size size fille with fill_value.
full_like	Returns a tensor with the same size as <pre>input</pre> fill_value.
quantize_per_tensor	Converts a float tensor to a quantize tensor with given scale an zero point.
quantize_per_channel	Converts a float tensor to a per-channel quantize tensor with given scales an zero points.
dequantize	Returns an fp32 Tensor y equantizing a quantize Tensor
complex	Constructs a complex tensor with its real part equal to real an its imaginary part equal to imag.
polar	Constructs a complex tensor whose elements are Cartesian coor inates correspon ing to the polar coor inates with a solute value abs an angle angle.
heaviside	Computes the Heavisi e step function for each element in input.
In exing, Slicing, Joining, Mutating O s	
adjoint	Returns a view of the tensor con ugate an with the last two imensions transpose .
argwhere	Returns a tensor containing the in ices of all non-zero elements of input.
an o timize vour ex erience, we serve cookies on this site. By clicking o	Concatenates the given sequence of tensors in tensors in the representation navigating, you agree to allow our usage of cookies. As the current maintains

Accordance Accordance Accordance in proceedings of the process of regard with a filippe standards. Accordance into the specific number of richards. Accordance into the specific number of richards. Sales injury, accordance into the specific number of richards. Concess accordance in y accordance into the specific number of richards. Concess accordance in y accordance in pricing the second in transit. Concess accordance in y accordance in pricing the second in transit. Concess accordance in the pricing the second in transit. Concess accordance in the pricing the second in an information and pricing the second in transit. Concess accordance in the pricing the second in an information and pricing the second in accordance in transit. Concess accordance in the pricing the second in accordance in accordance in the second in accordance in ac		
Altergrate speller reason lated to specifie seam or of class his Optit Tight opps, an error with show a manual invasioning first multiple central options common invasioning first multiple central options of process or region understand and common invasioning first multiple central options (deep did mink). Optic Light, an error with order or manual invasions in manual invasions in the central central options (deep did mink). Optic Light, an error with order or manual invasions in manual invasions in the central central optic central	concatenate	Alias of torch.cat().
Counts a required process of photose according to controlling and controlling	conj	Returns a view of input with a flippe con ugate it.
colors that a control gradual according to server in colors that according to server in colors that a color of the colors of the	chunk	Attempts to split a tensor into the specifie num er of chunks.
Stack propose appeared appeared appeared appeared appeared printing of the control of the contro	dsplit	
Carbon school along an acceptable by your. Carbon school along an acceptable by your. Splits based, acceptably accept high investor or restriction. Stock proper in paperoce for for stelly (calumn wine). See Trainer and () for function contription. See Trainer and () for function contribution contribution and () for see Trainer. See Trainer and () for function contribution contribution contribution. See Trainer and () for function contribution contribution and () for see Trainer. See Trainer and () for function contribution contribution contribution. See Trainer and () for function contribution contribu	column_stack	· · · · · · · · · · · · · · · · · · ·
Splits between a terms of the original between specimens and the production of the prod	dstack	Stack tensors in sequence epthwise (along thir axis).
transit transit social large or anticog, are established. Index add See Invest well () for function recorption. Index casy See Invest well () for function exception. Index casy See Invest well () for function exception. Index casy See Invest well () for function exception. Index casy See Invest well () for function exception. Index casy See Invest well () for function exception. Index well of See Invest well () for function exception. Index well of See Invest well () for function exception. Petern as a new Y-D tensor which in exec the Input tensor according to the orders make used which is a Bend Fence. More that interest data is a near over version of Input tensor. Petern as new Version data is a near over version of Input tensor. Petern as new Version data is a near over version of Input tensor. Petern as new Version data is a near over version of Input tensor. Petern as new Version data is a near over version of Input tensor. Petern as new Version data is a near over version of Input tensor. Petern as new Version data is a near over version of Input tensor. Petern as new Version data is a near over version of Input tensor. Petern as new Version data is a near very version of Input tensor of Input tensor. Petern as new Version of the original review version of Input tensor. Petern as new Version of the original review version of Input tensor and the record of Input tensor of Input tensor included and the Input tensor data is a Copy (officer data). Petern as new Version of the act tensor into Input tensor and the price in exception of Input tensor data is a Copy (officer data). Petern as new Version of the act tensor into Input tensor data is a Copy (officer data). Petern as new Version of the act tensor into Input tensor data is a Copy (officer data). Petern as new Version of the act tensor into Input tensor data is a Copy (officer data). Petern as new Version of the act tensor into Input tensor tensor into Input tensor. Petern as new Version of the act tensor into Input tensor tenso	gather	Gathers values along an axis specifie y dim.
See Index_set_() for function excription. Returns a new Violence of the Index_set index_	hsplit	
See sinday_ed_() for function excription. Index_relates See sinday_ed_() for function excription. Index_relates Recurss a new tensor which in exest the separt tensor along innersion all using the certains in index which is a long/resco. Tacked_solvet Recurss a new 1-D tensor which in exest the sees the solve tensor according to relate and seed to the sees the solve tensor. Moves the innersion(g) of separt acide position(g) in source to the position(g) in source to the position(g) in device tensor. The control of the second of the position(g) in source to the position(g) in source to the position(g) in device tensor. The control of the position(g) in device tensor. The control of the position(g) in source to the position of the po	hstack	Stack tensors in sequence horizontally (column wise).
See index_sedere_() for function escription. ### Returns a new tensor which in except to from tensor along immedian its using the entire in index which is a Long-Pennor. #### Returns a new 1-Diseasor which in except to story it ensure according to the colean mask state which is a Congressor. ###################################	index_add	See <pre>index_add_()</pre> for function escription.
Returns a new tensor which in exes the insust tensor along invested, salacit Returns a new tensor which in exes the insust tensor according to the colean mask assess which is a flority frence. Returns a new 1-0 tensor which in exes the insust tensor according to the colean mask assess which is a flority frence. Moves the invention(s) of insus at the position(s) in source to the position(s) in destination. Returns a new tensor that is a narrowe version of insust tensor. Returns a new tensor that is a narrowe version of insust tensor. Returns a new tensor with the same at an number of elements as about, wit with the specific shape. Returns a tensor with the same at an number of elements as about, wit with the specific shape. Returns a tensor with the same at an number of elements as about, wit with the specific shape. Returns a tensor with the same at an number of elements as about, wit with the specific shape. Returns a tensor with the same at an number of elements as about, with the specific shape. Returns a tensor with the same at an number of elements as about, with the specific shape. Returns a tensor with the same at an number of elements as about, with the specific shape. Returns a tensor with the same at an number of elements as about, with the specific shape. Returns a tensor with the same at an number of elements as about, with the specific shape. Returns a tensor with the same at an number of elements as about, with the specific shape. Returns a tensor with the same at an number of elements as about, and the specific shape. Returns a new tensor with the same at an number of elements as about, and the specific shape. Returns a new tensor with the same at an number of elements as about, and the specific shape. Returns a new tensor with the same at an number of elements as about the specific shape. Returns a new tensor with the same at an number of elements as about the specific shape. Returns a new tensor with the same at an number of elements as about the same at the position of sam	index_copy	See <pre>index_add_()</pre> for function escription.
invested, select Recurrs a new 1-D tensor which in execution is a Long Fenzor. Recurrs a new 1-D tensor which in execution and in select tensor tensor in adda which is a Boot Fenzor. Recurrs a new 1-D tensor which in execution (c) in source to the position(c) in destrication. Recurrs a new tensor that is a narrow: version of Liquit tensor. Recurrs a new tensor that is a narrow: version of Liquit tensor. Recurrs a new tensor that is a narrow: version of Liquit tensor. Same as Tensor, narrow: except this returns a copy rather than state storage. Recurrs a view of the original tensor tapps with its Timendans permute. Recurrs a view of the original tensor tapps with its Timendans permute. Recurrs a tensor with the same at an in num er of elements as Input; ut with the specific shape. Insultance Alias of teach, vetack(). Silices the taput tensor along the selecter imension at the given in ex. scorter Cut-of-place version of tench, Tansor, scatter. () Em e a the values of the are tensor into Liquit at the given in ex. Em e a the values of the are tensor into Liquit at the given in ex. Em e a the values of the are tensor into Liquit at the given in ex. Em e a the values of the are tensor into Liquit at the given in ex. Em e a the values of the are tensor into Liquit at the given in ex.	index_reduce	See <pre>index_reduce_() for function escription.</pre>
according to the colean mask resis which is a Bool tensor. Moves the imension(s) of input at the position(s) in source to the position(s) in destination. Alias for teach invedint). Returns a new tensor that is a narrow eversion of input tensor. Same as Tensor Astron() except this returns a copy rather than share storage. Returns a view of the original tensor input with its imensions permute. Returns a view of the original tensor input with its imensions permute. Returns a tensor with the same at an inum er of elements as input, ut with the specifie shape. Zon_stack Alias of teach_vstack(). Solices the input tensor along the selecter imension at the given in ex. Solices the values of the arc tensor into input along the electer in ex. Solices the values of the arc tensor into input along the select. Seatter Em e a the values of the arc tensor into input at the given in ex. Solice scatter Em e a the values of the arc tensor into input at the given in ex. Em e a the values of the arc tensor into input at the given in ex. Cut-of-place version of tooch lonear acceptance and () Cut-of-place version of tooch lonear acceptance and () Cut-of-place version of tooch lonear acceptance and ()	index_select	
moveaxis Alias for tarch-movedisc(). Returns a new tensor that is a narrowe version of input tensor. Same as lonear narrow() except this returns a copy rather than share storage. Returns a view of the original tensor lapus with its imensions permute. Returns a tensor with the same atta an num er of elements as imput, ut with the specifie shape. Tow_stack Alias of tarch_vatack(). Silices the input tensor along the selecte limension at the given in ex. Cut-of-place version of teach. Tensor_scatter_() allegonal_scatter Em e s the values of the sac tensor into input at the given in ex. Let us the same atta an num er of elements as imput, ut with the specifie shape. Em e s the values of the sac tensor into input at the given in ex. Let us show the sac tensor into input at the given in ex. Let us show the sac tensor into input at the given imension. Cut-of-place version of teach. Tensor_scatter_add_() Cut-of-place version of teach. Tensor_scatter_add_() Cut-of-place version of teach. Tensor_scatter_add_()	masked_select	
Returns a new tensor that is a narrowe version of input tensor. Same as Tensor narrow() except this returns a copy rather than share storage. Returns a view of the original tensor input with its imensions permute. Returns a view of the original tensor input with its imensions permute. Returns a tensor with the same atta an num er of elements as input, ut with the specific shape. Zon stack Alias of tozeh.vstack(). Silces the input tensor along the selecte imension at the given in ex. Cut-of-place version of toxch.Tensor.scatter_() that a she values of the size tensor into input along the input selects of input, with respect to dain an direct in ex. Em e she values of the size tensor into input at the given in ex. Em e she values of the size tensor into input at the given in ex. Cut-of-place version of toxch.Tensor.scatter_add_() Cut-of-place version of toxch.Tensor.scatter_add_()	movedim	
Same as Tonsor.narrow() except this returns a copy rather than share storage. Demute Returns a view of the original tensor input with its imensions permute. Returns a tensor with the same ata an num er of elements as input, ut with the specific shape. Tou.stack Alias of torch.vatack(). Slices the input tensor along the selecte imension at the given in ex. Cut-of-place version of torch.Tensor.scatter_() Em e s the values of the axe tensor into input along the iagonal elements of input, with respect to dint an dint. Em e s the values of the axe tensor into input at the given in ex. Let us the values of the axe tensor into input at the given in ex. Cut-of-place version of torch.Tensor.scatter_add_() Cut-of-place version of torch.Tensor.scatter_add_()	moveaxis	Alias for torch.movedim().
nazzou_copy nonzero Returns a view of the original tensor input with its imensions permute . Returns a tensor with the same at an num er of elements as input, ut with the specifie shape. row_stack Alias of torch_vstack() . Slices the input tensor along the selecte imension at the given in ex. scattor Out-of-place version of torch_Tensor_scatter_() Em e s the values of the src tensor into input along the ingonal elements of input, with respect to dind an din2 . Em e s the values of the src tensor into input at the given in ex. Slice_scatter Em e s the values of the src tensor into input at the given in ex. Cut-of-place version of torch_Tensor_scatter_add_() Cut-of-place version of torch_Tensor_scatter_add_()	narrow	Returns a new tensor that is a narrowe version of input tensor.
Returns a view of the original tensor input with its imensions permute . Returns a tensor with the same ata an num er of elements as input, ut with the specific shape. Returns a tensor with the same ata an num er of elements as input, ut with the specific shape. Alias of touch.vstack() . Slices the input tensor along the selecte imension at the given in ex. Out-of-place version of touch.Tensor.scatter_() Em e s the values of the size tensor into input along the iagonal elements of input, with respect to daint an diad . Em e s the values of the size tensor into input at the given in ex. Slices scatter Em e s the values of the size tensor into input at the given in ex. Out-of-place version of touch.Tensor.scatter_add_()	narrow_copy	
Returns a tensor with the same ata an num er of elements as input, ut with the specific shape. Tow_stack Alias of torch.vstack(). Slices the input tensor along the selecte imension at the given in ex. Cut-of-place version of torch.Tensor.scatter_() Em e s the values of the src tensor into input along the iagonal elements of input, with respect to dini an din2. Em e s the values of the src tensor into input at the given in ex. Em e s the values of the src tensor into input at the given in ex. Cut-of-place version of torch.Tensor.scatter_add_() Cut-of-place version of torch.Tensor.scatter_add_()	nonzero	
reshape input, ut with the specifie shape. Alias of torch.vstack(). Slices the input tensor along the selecte imension at the given in ex. Out-of-place version of torch.Tensor.scatter_() Em e s the values of the src tensor into input along the iagonal elements of input, with respect to dim1 and dim2. Em e s the values of the src tensor into input at the given in ex. Em e s the values of the src tensor into input at the given in ex. Cut-of-place version of torch.Tensor.scatter_add_() Out-of-place version of torch.Tensor.scatter_add_()	permute	
Slices the input tensor along the selecte imension at the given in ex. Substitution of touch. Tensor. scatter_() Em e s the values of the src tensor into input along the iagonal elements of input, with respect to dim1 and dim2. Em e s the values of the src tensor into input at the given in ex. Em e s the values of the src tensor into input at the given in ex. Em e s the values of the src tensor into input at the given in ex. Cut-of-place version of touch. Tensor. scatter_add_() Out-of-place version of touch. Tensor. scatter_add_()	reshape	
scatter Out-of-place version of torch.Tensor.scatter_() Em e s the values of the src tensor into input along the iagonal elements of input, with respect to dim1 an dim2. Em e s the values of the src tensor into input at the given in ex. Em e s the values of the src tensor into input at the given in ex. Em e s the values of the src tensor into input at the given imension. Out-of-place version of torch.Tensor.scatter_add_()	row_stack	Alias of torch.vstack().
Em e s the values of the src tensor into input along the iagonal elements of input, with respect to dim1 an dim2. Em e s the values of the src tensor into input at the given in ex. Em e s the values of the src tensor into input at the given in ex. Em e s the values of the src tensor into input at the given imension. Out-of-place version of torch. Tensor. scatter_add_()	select	
iagonal_scatter Em e s the values of the src tensor into input at the given in ex. Em e s the values of the src tensor into input at the given in ex. Em e s the values of the src tensor into input at the given imension. Out-of-place version of torch.Tensor.scatter_add_()	scatter	Out-of-place version of torch.Tensor.scatter_()
select_scatter in ex. Em e s the values of the src tensor into input at the given imension. Out-of-place version of torch.Tensor.scatter_add_() Out-of-place version of torch.Tensor.scatter_add_()	diagonal_scatter	
slice_scatter imension. Out-of-place version of torch.Tensor.scatter_add_() Out-of-place version of torch.Tensor.scatter_add_()	select_scatter	
Out-of-place version of torch Tensor scatter reduce ()	slice_scatter	The state of the s
Scatter_reduce Out-of-place version of torch.Tensor.scatter_reduce_()	scatter_add	Out-of-place version of torch.Tensor.scatter_add_()
	scatter_reduce	Out-of-place version of torch.Tensor.scatter_reduce_()

concat

squeeze	Returns a tensor with all specifie imensions of input of size 1 remove .
stack	Concatenates a sequence of tensors along a new imension.
swapaxes	Alias for torch.transpose().
swapdims	Alias for torch.transpose().
t	Expects input to e <= 2-D tensor an transposes imensions 0 an 1.
take	Returns a new tensor with the elements of input at the given in ices.
take_along_dim	Selects values from input at the 1- imensional in ices from indices along the given dim.
tensor_split	Splits a tensor into multiple su -tensors, all of which are views of input, along imension dim accor ing to the in ices or num er of sections specifie y indices_or_sections.
tile	Constructs a tensor y repeating the elements of input.
transpose	Returns a tensor that is a transpose version of input.
unbind	Removes a tensor imension.
unravel_index	Converts a tensor of flat in ices into a tuple of coor inate tensors that in ex into an ar itrary tensor of the specifie shape.
unsqueeze	Returns a new tensor with a imension of size one inserte at the specifie position.
vsplit	Splits input, a tensor with two or more imensions, into multiple tensors vertically accor ing to indices_or_sections.
vstack	Stack tensors in sequence vertically (row wise).
where	Return a tensor of elements selecte from either input or other, epen ing on condition.

Accelerators

Within the PyTorch repo, we efine an "Accelerator" as a torch.device that is eing use alongsi e a CP to spee up computation. These evice use an asynchronous execution scheme, using torch.Stream an torch.Event as their main way to perform synchronization. We also assume that only one such accelerator can e availa le at once on a given host. This allows us to use the current accelerator as the efault evice for relevant concepts such as pinne memory, Stream evice_type, FSDP, etc.

As of to ay, accelerator evices are (in no particular or er) "C DA", "MTIA", "XP", an Private se1 (many evice not in the PyTorch repo itself).

Stream	An in-or er queue of executing the respective tasks asynchronously in first in first out (FIFO) or er.
Event	Query an recor Stream status to i entify or control epen encies across Stream an measure timing.

Generators

Generator Generator Creates an returns a generator o ect that manages the state of the algorithm which pro uces pseu o ran om num ers.	Generator	Creates an returns a generator o ect that manages the state of the algorithm which pro uces pseu o ran om num ers.
--	-----------	--

Ran om sam ling

seed	Sets the see for generating ran om num ers to a non-eterministic ran om num er on all evices.
manual_seed	Sets the see for generating ran om num ers on all evices.
<pre>initial_seed</pre>	Returns the initial see for generating ran om num ers as a Python long.
get_rng_state	Returns the ran om num er generator state as a torch.ByteTensor.
set_rng_state	Sets the ran om num er generator state.

bernoulli	Draws inary ran om num ers (0 or 1) from a Bernoulli istri ution.
multinomial	Returns a tensor where each row contains num_samples in ices sample from the multinomial (a stricter efinition woul e multivariate, refer to torch.distributions.multinomial.Multinomial for more etails) pro a ility istri ution locate in the correspon ing row of tensor input.
normal	Returns a tensor of ran om num ers rawn from separate normal istri utions whose mean an stan ar eviation are given.
poisson	Returns a tensor of the same size as input with each element sample from a Poisson istri ution with rate parameter given y the correspon ing element in input i.e.,
rand	Returns a tensor fille $$ with ran $$ om num $$ ers from a uniform $$ istriuution on the interval $[0,1)$
rand_like	Returns a tensor with the same size as $rac{input}{n}$ that is fille with ran om num ers from a uniform istri ution on the interval $[0,1).$
randint	Returns a tensor fille with ran om integers generate uniformly etween low (inclusive) an high (exclusive).
randint_like	Returns a tensor with the same shape as Tensor input fille with ran om integers generate uniformly etween low (inclusive) an high (exclusive).
randn	Returns a tensor fille with ran om num ers from a normal istri ution with mean o an variance 1 (also calle the stan ar normal istri ution).
randn_like	Returns a tensor with the same size as input that is fille with ran om num ers from a normal istri ution with mean 0 an variance 1.
randperm	Returns a ran om permutation of integers from 0 to n - 1.

In- lace ran om sam ling

There are a few more in-place ran om sampling functions efine on Tensors as well. Click through to refer to their ocumentation:

- torch.Tensor.bernoulli_() in-place version of torch.bernoulli()
- torch.Tensor.cauchy_() num ers rawn from the Cauchy istri ution
- torch.Tensor.exponential_() num ers rawn from the exponential istri ution
- torch.Tensor.geometric_() elements rawn from the geometric istri ution
- torch.Tensor.log_normal_() samples from the log-normal istri ution
- torch.Tensor.normal_() in-place version of torch.normal()
- torch.Tensor.random_() num ers sample from the iscrete uniform istri ution
- torch.Tensor.uniform_() num ers sample from the continuous uniform istri ution

Quasi-ran om sam ling

quasirandom.SobolEngine is an engine for generating (scram le) So ol sequences.
--

Serialization

save	Saves an o ect to a isk file.
load	Loa s an o ect save with torch.save() from a file.

Parallelism

get_num_threads	Returns the num er of threa s use for parallelizing CP operations
set_num_threads	Sets the num er of threa s use for intraop parallelism on CP .
get_num_interop_threads	Returns the num er of threa s use for inter-op parallelism on CP (e.g.
set_num_interop_threads	Sets the num er of threa s use for interop parallelism (e.g.

Locally is a ling gra ient com utation

The context managers torch.no_grad(), torch.enable_grad(), an torch.set_grad_enabled() are helpful for locally is a ling an ena ling gra ient computation. See Locally is ling gra ient computation for more etails on their usage. These context managers are

```
>>> x = torch.zeros(1, requires_grad=True)
>>> with torch.no_grad():
y = x * 2
>>> y.requires_grad
False
>>> is_train = False
>>> with torch.set_grad_enabled(is_train):
... y = x * 2
>>> y.requires_grad
False
>>> torch.set_grad_enabled(True) # this can also be used as a function
>>> y = x * 2
>>> y.requires_grad
True
>>> torch.set_grad_enabled(False)
>>> y = x * 2
>>> y.requires_grad
False
```

no_grad	Context-manager that isa les gra ient calculation.
enable_grad	Context-manager that ena les gra ient calculation.
<pre>autograd.grad_mode.set_grad_enabled</pre>	Context-manager that sets gra ient calculation on or off.
is_grad_enabled	Returns True if gra mo e is currently ena le .
<pre>autograd.grad_mode.inference_mode</pre>	Context-manager that enalles or isalles inference moe.
<pre>is_inference_mode_enabled</pre>	Returns True if inference mo e is currently ena le .

Math o erations

Constants

inf	A floating-point positive infinity. Alias for math.inf.
nan	A floating-point "not a num er" value. This value is not a legal num er. Alias for math.nan.

Pointwise O s

Pointwise O s	
abs	Computes the a solute value of each element in input.
absolute	Alias for torch.abs()
acos	Computes the inverse cosine of each element in input.
arccos	Alias for torch.acos().
acosh	Returns a new tensor with the inverse hyper olic cosine of the elements of input.
arccosh	Alias for torch.acosh().
add	A s other, scale y alpha, to input.
addcdiv	Performs the element-wise ivision of tensor1 y tensor2, multiplies the result y the scalar value an a s it to input.
addcmul	Performs the element-wise multiplication of tensor1 y tensor2, multiplies the result y the scalar value an a s it to input.
angle	Computes the element-wise angle (in ra ians) of the given input tensor.
asin	Returns a new tensor with the arcsine of the elements of input.
arcsin	Alias for torch.asin().
asinh	Returns a new tensor with the inverse hyper olic sine of the elements of input.
arcsinh	Alias for torch.asinh().

Returns a new tensor with the arctangent of the elements of

atanh	Returns a new tensor with the inverse hyper olic tangent of the elements of input.
arctanh	Alias for torch.atanh().
atan2	Element-wise arctangent of $\mathrm{input}_i/\mathrm{other}_i$ with consignation of the quagrant.
arctan2	Alias for torch.atan2().
bitwise_not	Computes the itwise NOT of the given input tensor.
bitwise_and	Computes the itwise AND of input an other.
bitwise_or	Computes the itwise OR of input an other.
bitwise_xor	Computes the itwise XOR of input an other.
bitwise_left_shift	Computes the left arithmetic shift of input y other its.
bitwise_right_shift	Computes the right arithmetic shift of input y other its.
ceil	Returns a new tensor with the ceil of the elements of input, the smallest integer greater than or equal to each element.
clamp	Clamps all elements in input into the range [min, max].
clip	Alias for torch.clamp().
conj_physical	Computes the element-wise con ugate of the given input tensor.
copysign	Create a new floating-point tensor with the magnitue of input an the sign of other, elementwise.
cos	Returns a new tensor with the cosine of the elements of input.
cosh	Returns a new tensor with the hyper olic cosine of the elements of input.
deg2rad	Returns a new tensor with each of the elements of input converte from angles in egrees to ra ians.
div	Divi es each element of the input input y the correspon ing element of other.
divide	Alias for torch.div().
digamma	Alias for torch.special.digamma().
erf	Alias for torch.special.erf().
erfc	Alias for torch.special.erfc().
erfinv	Alias for torch.special.erfinv().
ехр	Returns a new tensor with the exponential of the elements of the input tensor input.
exp2	Alias for torch.special.exp2().
expm1	Alias for torch.special.expm1().
<pre>fake_quantize_per_channel_affine</pre>	Returns a new tensor with the ata in input fake quantize per channel using scale, zero_point, quant_min an quant_max, across the channel specifie y axis.
<pre>fake_quantize_per_tensor_affine</pre>	Returns a new tensor with the ata in input fake quantize using scale, zero_point, quant_min an quant_max.

float_power	Raises input to the power of exponent, elementwise, in ou le precision.
floor	Returns a new tensor with the floor of the elements of input, the largest integer less than or equal to each element.
floor_divide	
fmod	Applies C++'s st ::fmo entrywise.
frac	Computes the fractional portion of each element in input.
frexp	Decomposes input into mantissa an exponent tensors such that $input = mantissa \times 2^{\rm exponent}.$
gradient	Estimates the gralient of a function $g:\mathbb{R}^n \to \mathbb{R}$ in one or more imensions using the secon-or-or-or-or-or-or-or-or-or-or-or-or-or-
imag	Returns a new tensor containing imaginary values of the self tensor.
ldexp	Multiplies input y 2 ** other.
lerp	Does a linear interpolation of two tensors start (given y input) an end ase on a scalar or tensor weight an returns the resulting out tensor.
lgamma	Computes the natural logarithm of the a solute value of the gamma function on input.
log	Returns a new tensor with the natural logarithm of the elements of input.
log10	Returns a new tensor with the logarithm to the ase 10 of the elements of input.
log1p	Returns a new tensor with the natural logarithm of (1 + input).
log2	Returns a new tensor with the logarithm to the ase 2 of the elements of input.
logaddexp	Logarithm of the sum of exponentiations of the inputs.
logaddexp2	Logarithm of the sum of exponentiations of the inputs in ase-2.
logical_and	Computes the element-wise logical AND of the given input tensors.
logical_not	Computes the element-wise logical NOT of the given input tensor.
logical_or	Computes the element-wise logical OR of the given input tensors.
logical_xor	Computes the element-wise logical XOR of the given input tensors.
logit	Alias for torch.special.logit().
hypot	Given the legs of a right triangle, return its hypotenuse.
iO	Alias for torch.special.i0().
igamma	Alias for torch.special.gammainc().
igammac	Alias for torch.special.gammaincc().
mul	Multiplies input y other. Alias for torch.mul().
multiply	Alias for torch special multigammaln()
	AURS IOE Torch special multidammaln()

	respectively.
neg	Returns a new tensor with the negative of the elements of input.
negative	Alias for torch.neg()
nextafter	Return the next floating-point value after input towar s other, elementwise.
polygamma	Alias for torch.special.polygamma().
positive	Returns input.
pow	Takes the power of each element in input with exponent an returns a tensor with the result.
quantized_batch_norm	Applies atch normalization on a 4D (NCHW) quantize tensor.
quantized_max_pool1d	Applies a 1D max pooling over an input quantize tensor compose of several input planes.
quantized_max_pool2d	Applies a 2D max pooling over an input quantize tensor compose of several input planes.
rad2deg	Returns a new tensor with each of the elements of input converte from angles in rai ans to egrees.
real	Returns a new tensor containing real values of the self tensor.
reciprocal	Returns a new tensor with the reciprocal of the elements of input
remainder	Computes Python's mo ulus operation entrywise.
round	Roun s elements of input to the nearest integer.
rsqrt	Returns a new tensor with the reciprocal of the square-root of each of the elements of input.
sigmoid	Alias for torch.special.expit().
sign	Returns a new tensor with the signs of the elements of input.
sgn	This function is an extension of torch.sign() to complex tensors.
signbit	Tests if each element of input has its sign it set or not.
sin	Returns a new tensor with the sine of the elements of input.
sinc	Alias for torch.special.sinc().
sinh	Returns a new tensor with the hyper olic sine of the elements of <pre>input</pre> .
softmax	Alias for torch.nn.functional.softmax().
sqrt	Returns a new tensor with the square-root of the elements of input.
square	Returns a new tensor with the square of the elements of input.
sub	Su tracts other, scale y alpha, from input.
subtract	Alias for torch.sub().
tan	Returns a new tensor with the tangent of the elements of input.
tanh	Returns a new tensor with the hyper olic tangent of the elements of input.
a timize valur ex erience we serve cookies on this site. By clicking or	navigating, you agree to allow our usage of cookies. As the current main

input with the values specifie y nan, posinf, an neginf,

trunc	Returns a new tensor with the truncate integer values of the elements of input.
xlogy	Alias for torch.special.xlogy().
Re uction O s	
argmax	Returns the in ices of the maximum value of all elements in the input tensor.
argmin	Returns the in ices of the minimum value(s) of the flattene tensor or along a imension
amax	Returns the maximum value of each slice of the input tensor in the given imension(s) dim.
amin	Returns the minimum value of each slice of the <pre>input</pre> tensor in the given imension(s) <pre>dim</pre> .
aminmax	Computes the minimum an maximum values of the input tensor.
all	Tests if all elements in input evaluate to True.
any	Tests if any element in input evaluates to <i>True</i> .
max	Returns the maximum value of all elements in the input tensor.
min	Returns the minimum value of all elements in the input tensor.
dist	Returns the p-norm of (input - other)
logsumexp	Returns the log of summe exponentials of each row of the input tensor in the given imension dim.
mean	
nanmean	Computes the mean of all <i>non-NaN</i> elements along the specifie imensions.
median	Returns the me ian of the values in input.
nanmedian	Returns the me ian of the values in input, ignoring NaN values.
mode	Returns a name tuple (values, indices) where values is the mo e value of each row of the input tensor in the given imension dim, i.e. a value which appears most often in that row, an indices is the in ex location of each mo e value foun .
norm	Returns the matrix norm or vector norm of a given tensor.
nansum	Returns the sum of all elements, treating Not a Num ers (NaNs) as zero.
prod	Returns the pro uct of all elements in the input tensor.
quantile	Computes the q-th quantiles of each row of the input tensor along the imension dim.
nanquantile	This is a variant of torch.quantile() that "ignores" NaN values, computing the quantiles q as if NaN values in input i not exist.
std	Calculates the stan ar eviation over the imensions specifie y
std_mean	Calculates the stan ar eviation an mean over the imensions specifie y dim.
sum	Returns the sum of all elements in the input tensor.
unique	Returns the unique elements of the input tensor.
unique_consecutive	Eliminates all ut the first element from every consecutive group of equivalent elements.

var_mean	Calculates the variance an mean over the imensions specifie y dim.
count_nonzero	Counts the num er of non-zero values in the tensor input along the given dim.
Com arison O s	
allclose	This function checks if input an other satisfy the con ition:
argsort	Returns the in ices that sort a tensor along a given imension in ascen ing or er y value.
eq	Computes element-wise equality
equal	True if two tensors have the same size an elements, False otherwise.
ge	Computes $\mathrm{input} \geq \mathrm{other}$ element-wise.
greater_equal	Alias for torch.ge().
gt	Computes $\mathrm{input} > \mathrm{other}$ element-wise.
greater	Alias for torch.gt().
isclose	Returns a new tensor with oolean elements representing if each element of input is "close" to the correspon ing element of other.
isfinite	Returns a new tensor with oolean elements representing if each element is <i>finite</i> or not.
isin	Tests if each element of elements is in test_elements.
isinf	Tests if each element of input is infinite (positive or negative infinity) or not.
isposinf	Tests if each element of input is positive infinity or not.
isneginf	Tests if each element of input is negative infinity or not.
isnan	Returns a new tensor with oolean elements representing if each element of input is NaN or not.
isreal	Returns a new tensor with oolean elements representing if each element of input is real-value or not.
kthvalue	Returns a name tuple (values, indices) where values is the k th smallest element of each row of the input tensor in the given imension dim.
le	Computes $\mathrm{input} \leq \mathrm{other}$ element-wise.
less_equal	Alias for torch.le().
1t	Computes $\mathrm{input} < \mathrm{other}$ element-wise.
less	Alias for torch.lt().
maximum	Computes the element-wise maximum of input an other.
minimum	Computes the element-wise minimum of input an other.
fmax	Computes the element-wise maximum of input an other.
fmin	Computes the element-wise minimum of input an other.
ne	Computes $\mathrm{input} eq \mathrm{other}$ element-wise.

sent seasons green recessors de server avec despite for insention sont le propriet avec despite de la propriet avec despite de la propriet avec de la propriet ave	sort	Sorts the elements of the input tensor along a given imension in ascen ing or er y value.
sector of Cost	topk	Returns the k largest elements of the given input tensor along a given imension.
Activations and analysis of the counter transform (CIF). Activations are sentenced and analysis of facilities and analysis of facilities. Activation and facilities are also an analysis of facilities. Activation and facilities are also analysis of facilities. Activation and facilities are also analysis of facilities. Activation and facilities are also analysis of facilities. Activate analysis	msort	Sorts the elements of the input tensor along its first imension in ascen ing or er y value.
Institution in Process short the Paulier Transform. Institutionise Literature, delate Literature, delat	S ectral O s	
Destination of function and function of function and function are supported by the foliar windows and function are supported by the foliar windows are supported by the foliar windows and functions are supported by the function of functions are supported by the functions are supported by the function of functions are supported by the function of functions are supported by the function of the functions are supported by the function a	stft	Short-time Fourier transform (STFT).
biodiss passes biodiss passes biodiss passes biodiss passes brownisted biodiss passes bi	istft	Inverse short time Fourier Transform.
Harming win ow function. Harming win ow function. Harming ow now function. Harming ow or function. Compares the National Development of which in owlength shadow, require an object parameter sets. Other O criations Stillows_At Returns a 1- inventional view of each input tensor with zero inventions. Stillows_At Returns a 2- inventional view of each input tensor with zero inventions. Stillows_At Returns a 3- inventional view of each input tensor with zero inventions. Count the frequency of each value in an array of non-negative into. Count the frequency of each value in an array of non-negative into. Executions Consider the standard tensors according to find a count of the county of each input tensor with zero into. Similar to be standard tensors Similar to be standard tensors according to finds counting committee of the county of each input tensor with zero into. Similar to be standard tensors of ut for shapes. Societies Similar to be standard tensors of ut for shapes. Societies Compares according to find the county of each value in the following proper of each value in the following energy of each energy of eac	bartlett_window	Bartlett win ow function.
Harn win on function. Computes the foliaire win low with win ow length states, longers and shape parameter levels. Cher O erations States 1,00 Returns a 2- intensional view of each input tensor with zero intensions. Attending the states 1,00 Returns a 3- intensional view of each input tensor with zero intensions. Count the frequency of each value in an array of non-negative ints. Count the frequency of each value in an array of non-negative ints. Count the frequency of each value in an array of non-negative ints. Crease a look ligoral matrix from proville cleasers. Decodes 1, tensors Break casts the given tensors according to free casing permantical forms the indicated parameters in the state of the	blackman_window	Blackman win ow function.
Computes the lister win on with win tow length wholes largeth an observer, wholes are continued in the lister win on with win tow length wholes largeth an observer acts. Cherr O enations Steast 10 Returns a 1- Immensional view of each input tensor with zero immedions. Steast 30 Returns a 2- Immensional view of each input tensor with zero immedions. Steast 30 Steast 30 Count the frequency of each value in an array of non-negative inte. Count the frequency of each value in an array of non-negative inte. Steast 30 Bros. casts the given tensors according to Bros. casting semantics are seen to the shape shoot. Steast 30 String to the stage shoot. String to the string the string shoot. String the string the string shoot. String to the string the string shoot. String to the string shoot. String the st	hamming_window	Hamming win ow function.
Deter O erations attent_id atte	hann_window	Hann win ow function.
at least_2d at le	kaiser_window	
attent,2d	Other O erations	
atleast_3d atleast_3d atleast_3d atleast_3d atleast_3d atleast_3d Count the frequency of each value in an array of non-negative incs. Count the frequency of each value in an array of non-negative incs. Count the frequency of each value in an array of non-negative incs. Create a lock lagonal matrix from provil e tensors. broadcast_taneors Broa casts the given tensors according to Broa casting semantics. Broadcast_taneors Similar to broadcast_tensors() ut for shapes. Cautesian_secol Cautesian_secol Cautesian_secol Count the frequency of each value in an array of non-negative incs. Similar to broadcast_tensors() ut for shapes. Similar to broadcast_tensors() ut for shapes. Cautesian_secol Cautesian_secol Cautesian_secol Cautesian_pro uct of the given sequence of tensors. Counted Co	atleast_1d	
bisecount Count the frequency of each value in an array of non-negative ints. Dicak_clag Create a lock ingonal matrix from provi e tensors. Broat casts the given tensors according to Broat costing semantics throutcast_tensors Broat casts the given tensors according to Broat costing semantics broadcast_tonsors Broat casts the given tensors according to Broat costing semantics broadcast_tonsors Broat casts the given tensors according to Broat costing semantics broadcast_thapes Similar to broadcast_tensors() ut for shapes. Bectures the inces of the uckets to which each value in the saput elongs, where the our aries of the uckets are set y boundaries. Cattesian_prod Do cartesian product of the given sequence of tensors. Consults Computes atchet the p-norm istance etween each pair of the two collections of row vectors. Computes atchet the p-norm istance etween each pair of the two collections of row vectors. Computes atchet the p-norm istance etween each pair of the two collections of row vectors. Computes atchet the p-norm istance etween each pair of the two collections of row vectors. Computes atchet the p-norm istance etween each pair of the two collections of row vectors. Computes atchet the p-norm istance etween each pair of the two collections of row vectors. Computes atchet the p-norm istance etween each pair of the two collections of row vectors. Computes atchet the p-norm istance etween each pair of the two collections of row vectors. Computes atchet the p-norm istance etween each pair of the vectors in intension estimated in the vectors in intension estimated in the vector in intensi	atleast_2d	
block_diag Create a lock ingonal matrix from provi e tensors. Broa casts the given tensors accor ing to Broa casting semantics broadcast_tensors Broa casts input to the shape shape. Broa casts input to the shape shape. Broadcast_tensors() ut for shapes. Castesian_prod Do cartesian pro uct of the uckets to which each value in the input clongs, where the oun aries of the uckets are set y boundaries. Computes atche the p-norm istance etween each pair of the two collections of row vectors. Computes atche the p-norm istance etween each pair of the two collections of row vectors. Compute com inations of length p of the given tensor. Compute com inations of length p of the given tensor. Controef Estimates the Pearson pro uct-moment correlation coefficient matrix of the varia les given y the input matrix, where rows are the varia les given y the input matrix, where rows are the varia les an columns are the o servations. Returns the cross pro uct of vectors in imension die of input operations. Returns the cross pro uct of vectors in imension die of input	atleast_3d	•
broadcast_tensors Broa casts the given tensors accor ing to Broa casting semantics broadcast_to Broa casts input to the shape shape. Similar to broadcast_tensors() ut for shapes. broadcast_shapes Similar to broadcast_tensors() ut for shapes. Beturns the in ices of the uckets to which each value in the input elongs, where the oun aries of the uckets are set y boundaries. Cartesian_prod Do cartesian pro uct of the given sequence of tensors. Computes atche the p-norm istance etween each pair of the two collections of row vectors. Compute com inations of length r of the given tensor. Compute com inations of length r of the given tensor. Coxtcoef Estimates the Pearson pro uct-moment correlation coefficient matrix of the varia les given y the Input matrix, where rows are the varia les an columns are the o servations. Cov Setimates the covariance matrix of the varia les given y the Input matrix, where rows are the varia les an columns are the o servations. Returns the cross pro uct of vectors in innersion dain of input	bincount	
broadcast_to Broa casts input to the shape shape. broadcast_shapes Similar to broadcast_tonsors() ut for shapes. Broa casts input to the shape shape. Similar to broadcast_tonsors() ut for shapes. Broa casts input to the shape shape. Returns the in ices of the uckets to which each value in the input elongs, where the oun aries of the uckets are set y boundaries. Cartesian_prod Do cartesian pro uct of the given sequence of tensors. Computes atche the p-norm istance etween each pair of the two collections of row vectors. Clone Returns a copy of input. Combinations Compute com inations of length r of the given tensor. Compute com inations of length r of the given tensor. Constructed Estimates the Pearson pro uct-moment correlation coefficient matrix of the varia les given y the input matrix, where rows are the varia les an columns are the o servations. Cov Returns the cross pro uct of vectors in imension din of input Returns the cross pro uct of vectors in imension din of input	block_diag	Create a lock iagonal matrix from provi e tensors.
broadcast_shapes Similar to broadcast_tensors() ut for shapes. Returns the in ices of the uckets to which each value in the input elongs, where the oun aries of the uckets are set y boundaries; cartesian_prod Do cartesian pro uct of the given sequence of tensors. Computes atche the p-norm istance etween each pair of the two collections of row vectors. clone Returns a copy of input. Combinations Compute com inations of length r of the given tensor. Estimates the Pearson pro uct-moment correlation coefficient matrix of the varia les given y the input matrix, where rows are the varia les an columns are the o servations. Estimates the covariance matrix of the varia les given y the input matrix, where rows are the varia les an columns are the o servations. Returns the cross pro uct of vectors in imension die of input	broadcast_tensors	Broa casts the given tensors accor ing to Broa casting semantics.
Returns the in ices of the uckets to which each value in the input elongs, where the oun aries of the uckets are set y boundaries. Cartesian prod Do cartesian pro uct of the given sequence of tensors. Computes atche the p-norm istance etween each pair of the two collections of row vectors. Clone Returns a copy of input. Compute com inations of length r of the given tensor. Estimates the Pearson pro uct-moment correlation coefficient matrix of the varia les given y the input matrix, where rows are the varia les an columns are the o servations. Estimates the covariance matrix of the varia les given y the input matrix, where rows are the varia les an columns are the o servations. Returns the cross pro uct of vectors in imension dim of input	broadcast_to	Broa casts input to the shape shape.
bucketize input elongs, where the oun aries of the uckets are set y boundaries. Do cartesian pro uct of the given sequence of tensors. Computes atche the p-norm istance etween each pair of the two collections of row vectors. Returns a copy of input. Compute com inations of length r of the given tensor. Compute com inations of length r of the given tensor. Estimates the Pearson pro uct-moment correlation coefficient matrix of the varia les given y the input matrix, where rows are the varia les an columns are the o servations. Estimates the covariance matrix of the varia les given y the input matrix, where rows are the varia les an columns are the o servations. Returns the cross pro uct of vectors in imension dian of input	broadcast_shapes	Similar to broadcast_tensors() ut for shapes.
Computes atche the p-norm istance etween each pair of the two collections of row vectors. Clone Returns a copy of input. Combinations Compute com inations of length r of the given tensor. Estimates the Pearson pro uct-moment correlation coefficient matrix of the varia les given r the varia les given r the varia les an columns are the o servations. Estimates the covariance matrix of the varia les given r the varia les an columns are the o servations. Returns the cross pro uct of vectors in imension dim of input	bucketize	input elongs, where the oun aries of the uckets are set y
clone Returns a copy of input. Combinations Compute comminations of length r of the given tensor. Estimates the Pearson promute-moment correlation coefficient matrix of the varialles given by the input matrix, where rows are the varialles an columns are the observations. Estimates the covariance matrix of the varialles given by the input matrix, where rows are the varialles an columns are the observations. Returns the cross promute of vectors in imension dim of input matrix.	cartesian_prod	Do cartesian pro uct of the given sequence of tensors.
Compute com inations of length r of the given tensor. Estimates the Pearson pro uct-moment correlation coefficient matrix of the varia les given y the input matrix, where rows are the varia les an columns are the o servations. Estimates the covariance matrix of the varia les given y the input matrix, where rows are the varia les an columns are the o servations. Returns the cross pro uct of vectors in imension dim of input	cdist	
Estimates the Pearson pro uct-moment correlation coefficient matrix of the varia les given y the input matrix, where rows are the varia les an columns are the o servations. Estimates the covariance matrix of the varia les given y the input matrix, where rows are the varia les an columns are the o servations. Returns the cross pro uct of vectors in imension dim of input	clone	Returns a copy of input.
matrix of the varia les given y the input matrix, where rows are the varia les an columns are the o servations. Estimates the covariance matrix of the varia les given y the input matrix, where rows are the varia les an columns are the o servations. Returns the cross pro uct of vectors in imension dim of input	combinations	Compute com $$ inations of length r of the given tensor.
input matrix, where rows are the varia les an columns are the o servations. Returns the cross pro uct of vectors in imension dim of input	corrcoef	matrix of the varia les given y the input matrix, where rows are
	cov	input matrix, where rows are the varia les an columns are the
cross an other.	cross	

cummin	Returns a name tuple (values, indices) where values is the cumulative minimum of elements of input in the imension dim.
cumprod	Returns the cumulative pro uct of elements of input in the imension dim.
cumsum	Returns the cumulative sum of elements of input in the imension dim.
diag	If input is a vector (1-D tensor), then returns a 2-D square tensor
diag_embed	Creates a tensor whose iagonals of certain 2D planes (specifie y dim1 an dim2) are fille y input.
diagflat	 If input is a vector (1-D tensor), then returns a 2-D square tensor
diagonal	Returns a partial view of input with the its iagonal elements with respect to dim1 an dim2 appen e as a imension at the en of the shape.
diff	Computes the n-th forwar ifference along the given imension.
einsum	Sums the pro uct of the elements of the input operands along imensions specifie using a notation ase on the Einstein summation convention.
flatten	Flattens input y reshaping it into a one- imensional tensor.
flip	Reverse the or er of an n-D tensor along given axis in ims.
fliplr	Flip tensor in the left/right irection, returning a new tensor.
flipud	Flip tensor in the up/own irection, returning a new tensor.
kron	Computes the Kronecker pro uct, enote $y \otimes$, of input an other.
rot90	Rotate an n-D tensor y 90 egrees in the plane specifie y ims axis.
gcd	Computes the element-wise greatest common ivisor (GCD) of input an other.
histc	Computes the histogram of a tensor.
histogram	Computes a histogram of the values in a tensor.
histogramdd	Computes a multi- imensional histogram of the values in a tensor.
meshgrid	Creates gris of coor inates specifie y the 1D inputs in attr:tensors.
lcm	Computes the element-wise least common multiple (LCM) of input an other.
logcumsumexp	Returns the logarithm of the cumulative summation of the exponentiation of elements of input in the imension dim.
ravel	Return a contiguous flattene tensor.
renorm	Returns a tensor where each su -tensor of input along imension dim is normalize such that the <i>p</i> -norm of the su -tensor is lower than the value maxnorm
repeat_interleave	Repeat elements of a tensor.
roll	Roll the tensor input along the given imension(s).
searchsorted	Fin the in ices from the <i>innermost</i> imension of sorted_sequence such that, if the correspon ing values in values were inserte efore the in ices, when sorte, the or er of the correspon ing <i>innermost</i> imension within sorted_sequence woul e preserve.

Recurs the over straight apriced are extended to assert or cold and and have been about the other recult water and all and the straight apriced are extended to a server or state of the straight apriced are extended to a server or state of the straight and the straight apriced are in the straight and the straigh	trace	Returns the sum of the elements of the iagonal of the input 2-D matrix.
ENLINES Control to approximate policy or dismostration control to the control to	tril	atch of matrices input, the other elements of the result tensor
Seturns a few of the upper bragatic part of a con- yield many where the fact two contains two contains the co	tril_indices	matrix in a 2- y-N Tensor, where the first row contains row coor inates of all in ices an the secon row contains column
contacts as 2-year Transposition of the Form constants row control column care reads. ***Columns**	triu	of matrices input, the other elements of the result tensor out
Construction Constructs a view of Least as a real tensor. Recurs a rick of Least as a complex tensor. Recurs a rick of Least as a complex tensor. Recurs a rick of Least as a complex tensor. Recurs a rick of Least as a complex tensor. Recurs a rick of Least as a complex tensor. Recurs a rick of Least as a complex tensor. Recurs a rick of Least as a complex tensor. Recurs a rick of They, the recurs a beautiful tensor and tensor with materialize conjugation if Least as complex tensor with materialize conjugation if Least as tensor with materialize conjugation in Least as tensor with materialize conjugation in the material tensor is a tensor multiplication of the material tensor is an extensor with materialized and tensor with the received and as a set to the conformation of the material tensor and tensor with tensor with tensor property and of materialized and tensor with tensor property and of materialized and tensor with tensor property and of materialized and tensor with tensor property and the tensor with tensor with tensor property and ten	triu_indices	matrix in a 2- y-N Tensor, where the first row contains row coor inates of all in ices an the secon row contains column
Returns a view of upon to a complete cerebr. Returns a refer of upon to a complete cerebr. Returns a refer of upon to a complete cerebr. Returns a refer of upon to a complete cerebr. Returns a refer of upon to a complete cerebr. Returns a refer of upon to a complete cerebr. Returns a refer of upon to the complete cerebr. Returns a refer to from other cerebral section. Returns a refer to from other cerebral section. Returns a refer to from the cerebral section of upon to the cerebral section. Returns a refer to from the cerebral section of the returns to upon to the cerebral section. Returns a refer to from the cerebral section of the matrices come in testing and testing, which are used a section of the matrices cerebral and restrict multiplication of the matrices cerebral and restrict multiplication of the matrices cerebral and restrict to the cerebral section of the matrices cerebral and restrict to the cerebral section of the matrices cerebral and restrict to the cerebral section of the matrices cerebral and restrict to the cerebral section of the matrices cerebral and restrict to the cerebral section of the matrices cerebral and restrict to the cerebral section of the matrices cerebral and restrict to the cerebral section of the matrices cerebral and restrict to the cerebral section of the cerebral and restrict to the cerebral section of	unflatten	
Results a view of liquid as a complex states. Results a new sensor with materialize con ugadoff input is consistent to conjugate if it is not to True, else results (appl.) Results a new sensor with materialize con ugadoff input is conjugate in the conjugate in conjugate	vander	Generates a Van ermon e matrix.
Returns a new sensor with materialize con uppoint if should is consistent with materialize consupption if should is consistent with materialize and uppoint if should is negative. Returns a new sensor with materialize negation if should is negative. It is set to Traw, whe returns should. BLAS an LAPACK O errations Benforms a new sensor with materialize negation if should is negative. It is set to Traw, whe returns should. Penforms a cost matrix multiplication of the matrix sentor in tested, will also the use as seep (all materix multiplication of the matrix sentor in tested. Penforms a matrix multiplication of the matrix sent an institute of the matrix sent and the vector was. Penforms a matrix multiplication of the matrix set and the vector was. Penforms a cost matrix from use of vectors vest an institute to the matrix set and the vector was. Penforms a cost matrix from use of matrices in tested an institute. Penforms a cost matrix from use of matrices store in sapple and matrix. Returns the matrix project of the AT 2-D testers. Computes the matrix point of the AT 2-D testers. Challesty inverse Computes the inverse of a complete Hermitian or real grammetric positive efficite matrix plane in Scholady ecomposition. Computes the solution of a system of inner equations with complete Hermitian or real grammetric positive efficite matrix plane in Scholady ecomposition. Computes the otipes use of two 1D tensors. Allow of touchwater(): Allow of touchwater(): Allow of touchwater():	view_as_real	Returns a view of input as a real tensor.
Returns a new tensor with materialize negation if input is negative. It is set to Your, else returns separe. Returns a new tensor with materialize negation if input is negative. It is set to Your, else returns separe. Returns a new tensor with materialize negation if input is negative it is set to Your, else returns separe. Returns a material material with a test of your distinction of the material separe in set and the second material separe in set of your multiplications get accomulate along the first innersion). Returns a material vector project of the material and in the vector well. Performs a material vector project of the material and in the vector well. Performs a material vector project of the material and in the vector well. Performs a material vector project of the material and in the vector well. Performs a set in material project of vectors well an invest and in a set of the material and in the vector well. Performs a set in material project of materices and in a set of the material and in the vector well. Performs a set in material project of materices poore in imput and material project of materices positive efficite materices. Computes the finement of a complete Hermitian or real symmetric positive efficite his given in Cholesky server. Computes the original project in positive efficite his given in Cholesky server. This is a low-level function for calling LAPACK's gent irrectly. Innex Computes the copposition for calling LAPACK's gent irrectly.	view_as_complex	Returns a view of input as a complex tensor.
BLAS an LAPACK O erations Performs a ach matrix-matrix pro uct of matrices dore in barrat ann barrat, with a re-local and step (bill matrix multiplications get accumulate along the first, imension). addren Performs a matrix multiplication of the matrices and annexis multiplication of the matrices and annexis multiplication of the matrix multiplication of the	resolve_conj	
Performs a matrix moutuplication of the matrix sect an matrix multiplication of matrix sect an matrix multiplication matrix sect an matrix multiplication of matrix sect an a sit to the matrix multiplication of matrix sect an a sit to the matrix multiplication of matrices in batchin an batchin. Deforms a matrix matrix matrix pro uct of matrices in batchin an batchin. Returns the matrix matrix pro uct of matrices score in input an matrix. Computes the inverse of a complex Hermitian or real symmetric positive—efinite matrix A or for atches of symmetric positive—efinite matrix A or for atches of symmetric positive—efinite matrix section of a system of linear equations with complex Hermitian or real symmetric positive—efinite lits given its Cholesky accomposition. Computes the output uct of two ID tensors. Detail Computes the output uct of two ID tensors. Alian for tourn, literals, sect.) Alian for tourn, literals, sect.)	resolve_neg	_
bacht an isotop, with a reluce of steep (all matrix multiplications get accumulate along the first limension). Berforms a matrix multiplication of the matrices multiplication of the matrices multiplication of the matrix sort an limension. Berforms a matrix vector project of the matrix sort an limension of the matrix sort and the vector vec. Berforms a matrix vector project of vectors vect an limension of the matrix sort and the vector vector vectors vect and vector and so to the matrix sort of vectors vect and vector and so to the matrix sort of vectors vect and vector and so to the matrix sort of vectors vect and vector and so to the matrix sort of matrices in the vector and sort matrix sort of matrices in the vector and sort matrix sort of matrices in the vector and sort matrix sort of matrices in the vector and sort of matrices in the vector and sort of the matrix project of matrices store in sort and sort of the sort of the sort of matrices in the vector and sort of the sort of matrices in the vector and sort of the sort of the sort of matrices in the vector and sort of the	BLAS an LAPACK O erations	
Performs a matrix-vector project of the matrix nation the vector vec. Performs the outer-project of vectors wed an investigation of the matrix input. Performs a atch matrix-matrix project of matrices in batch1 an batch2. Performs a atch matrix-matrix project of matrices in batch1 an batch2. Performs a atch matrix-matrix project of matrices store in Lipput an invat2. Chalan method Returns the matrix project of the LN 2-D tensors. Computes the Cholesky ecomposition of a symmetric positive-efinite matrices. Computes the inverse of a complex Hermitian or real symmetric positive-efinite matrix A or for atches of symmetric positive-efinite matrix given its Cholesky ecomposition. Computes the inverse of a complex Hermitian or real symmetric positive-efinite matrix given its Cholesky ecomposition. Computes the solution of a system of linear equations with complex Hermitian or real symmetric positive-efinite lhs given its Cholesky ecomposition. Computes the ot project of two 1D tensors. Alian of toreth-outer(). Alian of toreth-outer().	addbmm	batch1 an batch2, with a re uce a step (all matrix
performs the outer-pro uct of vectors wed an vec2 an a sit to the matrix inpurt. Deforms a atch matrix-matrix pro uct of matrices in batch1 an batch2.	addmm	Performs a matrix multiplication of the matrices mat1 an mat2.
baddbam Performs a atch matrix-matrix pro uct of matrices in batch2 an batch2. Performs a atch matrix-matrix pro uct of matrices in batch3 an batch2. Performs a atch matrix-matrix pro uct of matrices store in input an mat2. Chain matual Returns the matrix pro uct of the N 2-D tensors. Computes the Cholesky ecomposition of a symmetric positive-efinite matrices. Computes the inverse of a complex Hermitian or real symmetric positive-efinite matrices. Computes the inverse of a complex Hermitian or real symmetric positive-efinite matrices given its Cholesky ecomposition. Computes the solution of a system of linear equations with complex Hermitian or real symmetric positive-efinite lhs given its Cholesky ecomposition. Computes the ot pro uct of two 1D tensors. This is a low-level function for calling LAPACK's geaff irectly. Alias of torch-outer(). Alias for torch-outer().	addmv	
born Performs a atch matrix-matrix pro uct of matrices store in input an mat2. Chain_matrol Returns the matrix pro uct of the N 2-D tensors. Computes the Cholesky ecomposition of a symmetric positive-efinite matrix A or for atches of symmetric positive-efinite matrices. Computes the inverse of a complex Hermitian or real symmetric positive-efinite matrices. Computes the inverse of a complex Hermitian or real symmetric positive-efinite matrices. Computes the solution of a system of linear equations with complex Hermitian or real symmetric positive-efinite linear equations with complex Hermitian or real symmetric positive-efinite linear equations with complex Hermitian or real symmetric positive-efinite linear equations with complex Hermitian or real symmetric positive-efinite linear equations with complex Hermitian or real symmetric positive-efinite linear equations with complex Hermitian or real symmetric positive-efinite linear equations with complex Hermitian or real symmetric positive-efinite linear equations with complex Hermitian or real symmetric positive-efinite linear equations with complex Hermitian or real symmetric positive-efinite linear equations with complex Hermitian or real symmetric positive-efinite linear equations with complex Hermitian or real symmetric positive-efinite linear equations with complex Hermitian or real symmetric positive-efinite linear equations with complex Hermitian or real symmetric positive-efinite linear equations with complex Hermitian or real symmetric positive-efinite linear equations with complex Hermitian or real symmetric positive-efinite linear equations with complex Hermitian or real symmetric positive-efinite linear equations with complex Hermitian or real symmetric positive-efinite linear equations with complex Hermitian or real symmetric positive-efinite linear equations with complex Hermitian or real symmetric positive-efinite linear equations with complex Hermitian or real symmetric positive-efinite linear equations with complex Hermitian equations wit	addr	
chain_matmul Returns the matrix pro_uct of the N 2-D tensors. Computes the Cholesky ecomposition of a symmetric positive- efinite matrix A or for atches of symmetric positive- efinite matrices. Computes the inverse of a complex Hermitian or real symmetric positive- efinite matrix given its Cholesky ecomposition. Computes the solution of a system of linear equations with complex Hermitian or real symmetric positive- efinite lhs given its Cholesky ecomposition. dot Computes the ot pro_uct of two 1D tensors. This is a low-level function for calling LAPACK's gearf irectly. ger Alias of tarch_outer(). Computes the ot pro_uct for 1D tensors.	baddbmm	
Computes the Cholesky ecomposition of a symmetric positive- efinite matrix A or for atches of symmetric positive- efinite matrices. Computes the inverse of a complex Hermitian or real symmetric positive- efinite matrix given its Cholesky ecomposition. Computes the solution of a system of linear equations with complex Hermitian or real symmetric positive- efinite lhs given its Cholesky ecomposition. Computes the ot pro uct of two 1D tensors. This is a low-level function for calling LAPACK's gearf irectly. gear Alias of torch outer().	bmm	·
efinite matrix A or for atches of symmetric positive- efinite matrices. Computes the inverse of a complex Hermitian or real symmetric positive- efinite matrix given its Cholesky ecomposition. Computes the solution of a system of linear equations with complex Hermitian or real symmetric positive- efinite lhs given its Cholesky ecomposition. Computes the solution of a system of linear equations with complex Hermitian or real symmetric positive- efinite lhs given its Cholesky ecomposition. Computes the ot pro uct of two 1D tensors. This is a low-level function for calling LAPACK's geqf irectly. ger Alias of torch.outer().	chain_matmul	Returns the matrix pro $\ \operatorname{uct}$ of the N 2-D tensors.
cholesky_inverse Computes the solution of a system of linear equations with complex Hermitian or real symmetric positive- efinite lhs given its Cholesky ecomposition. Computes the ot pro uct of two 1D tensors. Computes the ot pro uct of two 1D tensors. This is a low-level function for calling LAPACK's geqrf irectly. Alias of torch.outer(). Computes the ot pro uct for 1D tensors.	cholesky	efinite matrix \boldsymbol{A} or for $% \boldsymbol{A}$ atches of symmetric positive- efinite
complex Hermitian or real symmetric positive- efinite lhs given its Cholesky ecomposition. Computes the ot pro uct of two 1D tensors. This is a low-level function for calling LAPACK's geqrf irectly. get Alias of torch.outer(). Computes the ot pro uct for 1D tensors.	cholesky_inverse	
geqrf This is a low-level function for calling LAPACK's geqrf irectly. Alias of torch.outer(). Computes the ot pro uct for 1D tensors.	cholesky_solve	complex Hermitian or real symmetric positive- efinite lhs given
ger Alias of torch.outer(). Computes the ot pro uct for 1D tensors. Alias for torch.linals.inv()	dot	Computes the ot pro uct of two 1D tensors.
inner Computes the ot pro uct for 1D tensors. Alias for torch_linalg.inv()	geqrf	This is a low-level function for calling LAPACK's geqrf irectly.
Alias for torch.linalg.inv()	ger	Alias of torch.outer().
<pre>inverse</pre> Alias for torch.linalg.inv()	inner	Computes the ot pro uct for 1D tensors.
	inverse	Alias for torch.linalg.inv()

logdet	Calculates log eterminant of a square matrix or atches of square matrices.
slogdet	Alias for torch.linalg.slogdet()
lu	Computes the L factorization of a matrix or atches of matrices A.
lu_solve	Returns the L solve of the linear system $Ax=b$ using the partially pivote L factorization of A from ${\tt lu_factor()}$.
lu_unpack	npacks the L ecomposition returne y lu_factor() into the <i>P, L, U</i> matrices.
matmul	Matrix pro uct of two tensors.
matrix_power	Alias for torch.linalg.matrix_power()
matrix_exp	Alias for torch.linalg.matrix_exp().
mm	Performs a matrix multiplication of the matrices input an mat2.
mv	Performs a matrix-vector pro uct of the matrix input an the vector vec.
orgqr	Alias for torch.linalg.householder_product().
ormqr	Computes the matrix-matrix multiplication of a pro uct of Househol er matrices with a general matrix.
outer	Outer pro uct of input an vec2.
pinverse	Alias for torch.linalg.pinv()
qr	Computes the QR ecomposition of a matrix or a atch of matrices input, an returns a name tuple (Q, R) of tensors such that $\operatorname{input} = QR$ with Q eing an orthogonal matrix or atch of orthogonal matrices an R eing an upper triangular matrix or atch of upper triangular matrices.
svd	Computes the singular value ecomposition of either a matrix or atch of matrices input.
svd_lowrank	Return the singular value ecomposition (U, S, V) of a matrix, atches of matrices, or a sparse matrix A such that $A pprox U \operatorname{diag}(S) V^{\mathrm{H}}.$
pca_lowrank	Performs linear Principal Component Analysis (PCA) on a low-rank matrix, atches of such matrices, or sparse matrix.
lobpcg	Fin the k largest (or smallest) eigenvalues an the correspon ing eigenvectors of a symmetric positive efinite generalize eigenvalue pro lem using matrix-free LOBPCG metho s.
trapz	Alias for torch.trapezoid().
trapezoid	Computes the trapezoi al rule along dim.
cumulative_trapezoid	Cumulatively computes the trapezoi al rule along dim.
triangular_solve	Solves a system of equations with a square upper or lower triangular invertigle matrix A an $\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \$
vdot	Computes the ot pro uct of two 1D vectors along a imension.
Foreach O erations	

Foreach O erations

_foreach_abs_

• WARNING		
This API is in eta an su ect to future changes. Forwar -mo e AD is not supporte .		
_foreach_abs	Apply torch.abs() to each Tensor of the input list.	
foreach abs	Apply torch.abs() to each Tensor of the input list.	

_foreach_acos_	Apply torch.acos() to each Tensor of the input list.
_foreach_asin	Apply torch.asin() to each Tensor of the input list.
_foreach_asin_	Apply torch.asin() to each Tensor of the input list.
_foreach_atan	Apply torch.atan() to each Tensor of the input list.
_foreach_atan_	Apply torch.atan() to each Tensor of the input list.
_foreach_ceil	Apply torch.ceil() to each Tensor of the input list.
_foreach_ceil_	Apply torch.ceil() to each Tensor of the input list.
_foreach_cos	Apply torch.cos() to each Tensor of the input list.
_foreach_cos_	Apply torch.cos() to each Tensor of the input list.
_foreach_cosh	Apply torch.cosh() to each Tensor of the input list.
_foreach_cosh_	Apply torch.cosh() to each Tensor of the input list.
_foreach_erf	Apply torch.erf() to each Tensor of the input list.
_foreach_erf_	Apply torch.erf() to each Tensor of the input list.
_foreach_erfc	Apply torch.erfc() to each Tensor of the input list.
_foreach_erfc_	Apply torch.erfc() to each Tensor of the input list.
_foreach_exp	Apply torch.exp() to each Tensor of the input list.
_foreach_exp_	Apply torch.exp() to each Tensor of the input list.
_foreach_expm1	Apply torch.expm1() to each Tensor of the input list.
_foreach_expm1_	Apply torch.expm1() to each Tensor of the input list.
_foreach_floor	Apply torch.floor() to each Tensor of the input list.
_foreach_floor_	Apply torch.floor() to each Tensor of the input list.
_foreach_log	Apply torch.log() to each Tensor of the input list.
_foreach_log_	Apply torch.log() to each Tensor of the input list.
_foreach_log10	Apply torch.log10() to each Tensor of the input list.
_foreach_log10_	Apply torch.log10() to each Tensor of the input list.
_foreach_log1p	Apply torch.log1p() to each Tensor of the input list.
_foreach_log1p_	Apply torch.log1p() to each Tensor of the input list.
_foreach_log2	Apply torch.log2() to each Tensor of the input list.
_foreach_log2_	Apply torch.log2() to each Tensor of the input list.
_foreach_neg	Apply torch.neg() to each Tensor of the input list.

_foreach_tan_ Ap	oply torch.tan() to each Tensor of the input list.
_foreach_sin	oply torch.sin() to each Tensor of the input list.
_foreach_sin_ Ap	oply torch.sin() to each Tensor of the input list.
_foreach_sinh	oply torch.sinh() to each Tensor of the input list.
_foreach_sinh_ Ap	oply torch.sinh() to each Tensor of the input list.
_foreach_round	oply torch.round() to each Tensor of the input list.
_foreach_round_ Ap	oply torch.round() to each Tensor of the input list.
_foreach_sqrt Ap	oply torch.sqrt() to each Tensor of the input list.
_foreach_sqrt_ Ap	oply torch.sqrt() to each Tensor of the input list.
_foreach_lgamma Ap	oply torch.lgamma() to each Tensor of the input list.
_foreach_lgamma_ Ap	oply torch.lgamma() to each Tensor of the input list.
_foreach_frac	oply torch.frac() to each Tensor of the input list.
_foreach_frac_ Ap	oply torch.frac() to each Tensor of the input list.
_foreach_reciprocal	oply torch.reciprocal() to each Tensor of the input list.
_foreach_reciprocal_	oply torch.reciprocal() to each Tensor of the input list.
_foreach_sigmoid Ap	oply torch.sigmoid() to each Tensor of the input list.
_foreach_sigmoid_	oply torch.sigmoid() to each Tensor of the input list.
_foreach_trunc Ap	oply torch.trunc() to each Tensor of the input list.
_foreach_trunc_	oply torch.trunc() to each Tensor of the input list.
_foreach_zero_	oply torch.zero() to each Tensor of the input list.
tilities	
	eturns whether PyTorch was uilt with GLIBCXX_ SE_CXX11_ABI=1
	eturns the torch.dtype that woul result from performing an ithmetic operation on the provi e input tensors.
	etermines if a type conversion is allowe un er PyTorch casting les escri e in the type promotion ocumentation.
	eturns the torch.dtype with the smallest size an scalar kin at is not smaller nor of lower kin than either type1 or type2.
	ets whether PyTorch operations must use "eterministic" gorithms.
are_deterministic_algorithms_enabled Re	eturns True if the glo al eterministic flag is turne on.
is_deterministic_algorithms_warn_only_enabled Re	eturns True if the glo al eterministic flag is set to warn only.
set_deterministic_debug_mode Se	ets the e ug mo e for eterministic operations.

set_float32_matmul_precision	Sets the internal precision of float32 matrix multiplications.	
get_float32_matmul_precision	Returns the current value of float32 matrix multiplication precision.	
set_warn_always	When this flag is False (efault) then some PyTorch warnings may only appear once per process.	
get_device_module	Returns the mo ule associate with a given evice(e.g., torch. evice('cu a'), "mtia:0", "xpu",).	
is_warn_always_enabled	Returns True if the glo al warn_always flag is turne on.	
vmap	vmap is the vectorizing map; vmap(func) returns a new function that maps func over some imension of the inputs.	
_assert	A wrapper aroun Python's assert which is sym olically tracea le.	
Sym olic Num ers		
CLASS torch.SymInt(node) [SO RCE]		
Like an int (inclu ing magic metho s), ut re irects all operations on the wrappe no e. This is use in particular to sym olically recor operations in the sym olic shape workflow.		
as_integer_ratio() [SO RCE]		
Represent this int as an exact integer ratio		

Return ty e

Tuple[SymInt, int]

```
CLASS torch.SymFloat(node) [SO RCE]
```

Like an float (inclu ing magic metho s), ut re irects all operations on the wrappe no e. This is use in particular to sym olically recor operations in the sym olic shape workflow.

```
as_integer_ratio() [SO RCE]
```

Represent this float as an exact integer ratio

Return ty e

Tuple[int, int]

```
con ugate() [SO RCE]
```

Returns the complex con ugate of the float.

Return ty e

SymFloat

```
hex() [SO RCE]
```

Returns the hexa ecimal representation of the float.

Return ty e

str

```
is_integer() [SO RCE]
```

Return True if the float is an integer.

```
CLASS torch.SymBool(node) [SO RCE]
```

Like an ool (inclu ing magic metho s), ut re irects all operations on the wrappe no e. This is use in particular to sym olically recor operations in the sym olic shape workflow.

nlike regular ools, regular oolean operators will force extra guar s instea of sym olically evaluate. se the itwise operators instea to han le this.

SymInt-aware utility for float casting. sym_float

sym_fresh_size

sym_sum

sym_int	SymInt-aware utility for int casting.
sym_max	SymInt-aware utility for max which avois ranching on a < .
sym_min	SymInt-aware utility for min().
sym_not	SymInt-aware utility for logical negation.
sym_ite	

inary a ition.

N-ary a which is faster to compute for long lists than iterate

• WARNING

This feature is a prototype an may have compatifility reaking changes in the future.

export generate /export /in ex

Control Flow

• WARNING

This feature is a prototype an may have compati ility reaking changes in the future.

Con itionally applies true_fn or false_fn.

O timizations

Optimizes given mo el/function using TorchDynamo an specifie acken .

torch.compile ocumentation

O erator Tags

CLASS torch.Tag Mem ers: core ata_ epen ent_output ynamic_output_shape flexi le_layout generate inplace_view may e_aliasing_or_mutating nee s_fixe _stri e_or er non eterministic_ itwise non eterministic_see e pointwise pt2_compliant_tag view_copy PROPERTY name

∠ Previous

© Co yright 2024, PyTorch Contri utors.

Built with S hinx using a theme rovi e y Rea the Docs.

Docs

Access com rehensive evelo er ocumentation for PyTorch

View Docs >

Tutorials

Get in- e th tutorials for eginners an ocumentation for PyTorch

a vance evelo ers

View Tutorials >

View Tutorials >

View Resources

View Resources >

PyTorch Po casts PyTorch Resources Stay u to ate Get Starte **Tutorials** Face ook S otify A le Features Docs Twitter Ecosystem Discuss YouTu e Google Blog Githu Issues Linke In Amazon Bran Gui elines Contri uting Privacy Terms

© Co yright The Linux Foun ation. The PyTorch Foun ation is a rolect of The Linux Foun ation. For we site terms of use, tral emark olicy an other olicies a licalle to The PyTorch Foun ation lease see www.linuxfoun ation.org/ olicies/. The PyTorch Foun ation su orts the PyTorch of en source rolect, which has een estalishe as PyTorch Project a Series of LF Projects, LLC. For olicies a licalle to the PyTorch Project a Series of LF Projects, LLC, lease see www.lf.nuxfoun.org/ olicies/.