torch.sparse

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 1)

Unknown directive type "automodule".

```
.. automodule:: torch.sparse
```

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 3)

Unknown directive type "currentmodule".

```
.. currentmodule:: torch
```

Introduction

PyTorch provides :class:`torch.Tensor` to represent a multi-dimensional array containing elements of a single data type. By default, array elements are stored contiguously in memory leading to efficient implementations of various array processing algorithms that relay on the fast access to array elements. However, there exists an important class of multi-dimensional arrays, so-called sparse arrays, where the contiguous memory storage of array elements turns out to be suboptimal. Sparse arrays have a property of having a vast portion of elements being equal to zero which means that a lot of memory as well as processor resources can be spared if only the non-zero elements are stored or/and processed. Various sparse storage formats (such as COO, CSR/CSC, LIL, etc.) have been developed that are optimized for a particular structure of non-zero elements in sparse arrays as well as for specific operations on the arrays.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 13); *backlink*

Unknown interpreted text role "class".

Note

When talking about storing only non-zero elements of a sparse array, the usage of adjective "non-zero" is not strict: one is allowed to store also zeros in the sparse array data structure. Hence, in the following, we use "specified elements" for those array elements that are actually stored. In addition, the unspecified elements are typically assumed to have zero value, but not only, hence we use the term "fill value" to denote such elements.

Note

Using a sparse storage format for storing sparse arrays can be advantageous only when the size and sparsity levels of arrays are high. Otherwise, for small-sized or low-sparsity arrays using the contiguous memory storage format is likely the most efficient approach.

Warning

The PyTorch API of sparse tensors is in beta and may change in the near future.

Sparse COO tensors

PyTorch implements the so-called Coordinate format, or COO format, as one of the storage formats for implementing sparse tensors. In COO format, the specified elements are stored as tuples of element indices and the corresponding values. In particular,

- the indices of specified elements are collected in `indices` tensor of size `(ndim, nse)` and with element type `torch.int64`,
- the corresponding values are collected in `values` tensor of size `(nse,)` and with an arbitrary integer or floating point number element type,

where `ndim` is the dimensionality of the tensor and `nse` is the number of specified elements.

Note

The memory consumption of a sparse COO tensor is at least `(ndim * 8 + <size of element type in bytes>) * nse` bytes (plus a constant overhead from storing other tensor data).

The memory consumption of a strided tensor is at least `product(<tensor shape>) * <size of element type in bytes>`.

For example, the memory consumption of a 10 000 x 10 000 tensor with 100 000 non-zero 32-bit floating point numbers is at least `(2 * 8 + 4) * 100 000 = 2 000 000` bytes when using COO tensor layout and `10 000 * 10 000 * 4 = 400 000 000` bytes when using the default strided tensor layout. Notice the 200 fold memory saving from using the COO storage format.

Construction

A sparse COO tensor can be constructed by providing the two tensors of indices and values, as well as the size of the sparse tensor (when it cannot be inferred from the indices and values tensors) to a function :func:`torch.sparse_coo_tensor`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 93); *backlink*

Unknown interpreted text role "func".

Suppose we want to define a sparse tensor with the entry 3 at location (0, 2), entry 4 at location (1, 0), and entry 5 at location (1, 2). Unspecified elements are assumed to have the same value, fill value, which is zero by default. We would then write:

```
>>> i = [[0, 1, 1],
         [2, 0, 2]]
>>> v =  [3, 4, 5]
>>> s = torch.sparse_coo_tensor(i, v, (2, 3))
>>> s
tensor(indices=tensor([[0, 1, 1],
                       [2, 0, 2]]),
       values=tensor([3, 4, 5]),
       size=(2, 3), nnz=3, layout=torch.sparse_coo)
>>> s.to_dense()
tensor([[0, 0, 3],
        [4, 0, 5]])
```

Note that the input `i` is NOT a list of index tuples. If you want to write your indices this way, you should transpose before passing them to the sparse constructor:

```
>>> i = [[0, 2], [1, 0], [1, 2]]
>>> v =  [3,      4,      5    ]
>>> s = torch.sparse_coo_tensor(list(zip(*i)), v, (2, 3))
>>> # Or another equivalent formulation to get s
>>> s = torch.sparse_coo_tensor(torch.tensor(i).t(), v, (2, 3))
>>> torch.sparse_coo_tensor(i.t(), v, torch.Size([2,3])).to_dense()
tensor([[0, 0, 3],
        [4, 0, 5]])
```

An empty sparse COO tensor can be constructed by specifying its size only:

```
>>> torch.sparse_coo_tensor(size=(2, 3))
tensor(indices=tensor([], size=(2, 0)),
       values=tensor([], size=(0,)),
       size=(2, 3), nnz=0, layout=torch.sparse_coo)
```

Hybrid sparse COO tensors

Pytorch implements an extension of sparse tensors with scalar values to sparse tensors with (contiguous) tensor values. Such tensors are called hybrid tensors.

PyTorch hybrid COO tensor extends the sparse COO tensor by allowing the `values` tensor to be a multi-dimensional tensor so that we have:

- the indices of specified elements are collected in `indices` tensor of size `(sparse_dims, nse)` and with element type `torch.int64`,
- the corresponding (tensor) values are collected in `values` tensor of size `(nse, dense_dims)` and with an arbitrary integer or floating point number element type.

Note

We use (M + K)-dimensional tensor to denote a N-dimensional hybrid sparse tensor, where M and K are the numbers of sparse and dense dimensions, respectively, such that M + K == N holds.

Suppose we want to create a (2 + 1)-dimensional tensor with the entry [3, 4] at location (0, 2), entry [5, 6] at location (1, 0), and entry [7, 8] at location (1, 2). We would write

```
>>> i = [[0, 1, 1],
         [2, 0, 2]]
>>> v =  [[3, 4], [5, 6], [7, 8]]
>>> s = torch.sparse_coo_tensor(i, v, (2, 3, 2))
>>> s
tensor(indices=tensor([[0, 1, 1],
                       [2, 0, 2]]),
       values=tensor([[3, 4],
                      [5, 6],
                      [7, 8]]),
       size=(2, 3, 2), nnz=3, layout=torch.sparse_coo)
```

```
>>> s.to_dense()
tensor([[[0, 0],
         [0, 0],
         [3, 4]],
        [[5, 6],
         [0, 0],
         [7, 8]]])
```

In general, if `s` is a sparse COO tensor and `M = s.sparse_dim()`, `K = s.dense_dim()`, then we have the following invariants:

- `M + K == len(s.shape) == s.ndim` - dimensionality of a tensor is the sum of the number of sparse and dense dimensions,
- `s.indices().shape == (M, nse)` - sparse indices are stored explicitly,
- `s.values().shape == (nse,) + s.shape[M : M + K]` - the values of a hybrid tensor are K-dimensional tensors,
- `s.values().layout == torch.strided` - values are stored as strided tensors.

Note

Dense dimensions always follow sparse dimensions, that is, mixing of dense and sparse dimensions is not supported.

Uncoalesced sparse COO tensors

PyTorch sparse COO tensor format permits *uncoalesced* sparse tensors, where there may be duplicate coordinates in the indices; in this case, the interpretation is that the value at that index is the sum of all duplicate value entries. For example, one can specify multiple values, `3` and `4`, for the same index `1`, that leads to an 1-D uncoalesced tensor:

```
>>> i = [[1, 1]]
>>> v =  [3, 4]
>>> s=torch.sparse_coo_tensor(i, v, (3,))
>>> s
tensor(indices=tensor([[1, 1]]),
       values=tensor(  [3, 4]),
       size=(3,), nnz=2, layout=torch.sparse_coo)
```

while the coalescing process will accumulate the multi-valued elements into a single value using summation:

```
>>> s.coalesce()
tensor(indices=tensor([[1]]),
       values=tensor([7]),
       size=(3,), nnz=1, layout=torch.sparse_coo)
```

In general, the output of :meth:`torch.Tensor.coalesce` method is a sparse tensor with the following properties:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 234); *backlink*

Unknown interpreted text role "meth".

- the indices of specified tensor elements are unique,
- the indices are sorted in lexicographical order,
- :meth:`torch.Tensor.is_coalesced()` returns `True`.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 239); *backlink*

Unknown interpreted text role "meth".

Note

For the most part, you shouldn't have to care whether or not a sparse tensor is coalesced or not, as most operations will work identically given a coalesced or uncoalesced sparse tensor.

However, some operations can be implemented more efficiently on uncoalesced tensors, and some on coalesced tensors.

For instance, addition of sparse COO tensors is implemented by simply concatenating the indices and values tensors:

```
>>> a = torch.sparse_coo_tensor([[1, 1]], [5, 6], (2,))
>>> b = torch.sparse_coo_tensor([[0, 0]], [7, 8], (2,))
>>> a + b
tensor(indices=tensor([[0, 0, 1, 1]]),
       values=tensor([7, 8, 5, 6]),
       size=(2,), nnz=4, layout=torch.sparse_coo)
```

If you repeatedly perform an operation that can produce duplicate entries (e.g., :func:`torch.Tensor.add`), you should occasionally coalesce your sparse tensors to prevent them from growing too large.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 260); *backlink*

Unknown interpreted text role "func".

On the other hand, the lexicographical ordering of indices can be advantageous for implementing algorithms that involve many element selection operations, such as slicing or matrix products.

Working with sparse COO tensors

Let's consider the following example:

```
>>> i = [[0, 1, 1],
         [2, 0, 2]]
>>> v =  [[3, 4], [5, 6], [7, 8]]
>>> s = torch.sparse_coo_tensor(i, v, (2, 3, 2))
```

As mentioned above, a sparse COO tensor is a :class:`torch.Tensor` instance and to distinguish it from the *Tensor* instances that use some other layout, on can use :attr:`torch.Tensor.is_sparse` or :attr:`torch.Tensor.layout` properties:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 279); *backlink*

Unknown interpreted text role "class".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 279); *backlink*

Unknown interpreted text role "attr".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 279); *backlink*

Unknown interpreted text role "attr".

```
>>> isinstance(s, torch.Tensor)
True
>>> s.is_sparse
True
>>> s.layout == torch.sparse_coo
True
```

The number of sparse and dense dimensions can be acquired using methods :meth:`torch.Tensor.sparse_dim` and :meth:`torch.Tensor.dense_dim`, respectively. For instance:

**System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst, line 291); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst, line 291); *backlink*

Unknown interpreted text role "meth".

```
>>> s.sparse_dim(), s.dense_dim()
(2, 1)
```

If `s` is a sparse COO tensor then its COO format data can be acquired using methods :meth:`torch.Tensor.indices()` and :meth:`torch.Tensor.values()`.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst, line 299); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst, line 299); *backlink*

Unknown interpreted text role "meth".

Note

Currently, one can acquire the COO format data only when the tensor instance is coalesced:

```
>>> s.indices()
RuntimeError: Cannot get indices on an uncoalesced tensor, please call .coalesce() first
```

For acquiring the COO format data of an uncoalesced tensor, use :func:`torch.Tensor._values()` and :func:`torch.Tensor._indices()`:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst, line 311); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst, line 311); *backlink*

Unknown interpreted text role "func".

```
>>> s._indices()
tensor([[0, 1, 1],
        [2, 0, 2]])
```

Warning

Calling :meth:`torch.Tensor._values()` will return a *detached* tensor. To track gradients, :meth:`torch.Tensor.coalesce().values()` must be used instead.

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst, line 321); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst, line 321); *backlink*

Unknown interpreted text role "meth".

Constructing a new sparse COO tensor results a tensor that is not coalesced:

```
>>> s.is_coalesced()
False
```

but one can construct a coalesced copy of a sparse COO tensor using the :meth:`torch.Tensor.coalesce` method:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 331); *backlink*

Unknown interpreted text role "meth".

```
>>> s2 = s.coalesce()
>>> s2.indices()
tensor([[0, 1, 1],
        [2, 0, 2]])
```

When working with uncoalesced sparse COO tensors, one must take into an account the additive nature of uncoalesced data: the values of the same indices are the terms of a sum that evaluation gives the value of the corresponding tensor element. For example, the scalar multiplication on an uncoalesced sparse tensor could be implemented by multiplying all the uncoalesced values with the scalar because `c * (a + b) == c * a + c * b` holds. However, any nonlinear operation, say, a square root, cannot be implemented by applying the operation to uncoalesced data because `sqrt(a + b) == sqrt(a) + sqrt(b)` does not hold in general.

Slicing (with positive step) of a sparse COO tensor is supported only for dense dimensions. Indexing is supported for both sparse and dense dimensions:

```
>>> s[1]
tensor(indices=tensor([[0, 2]]),
       values=tensor([[5, 6],
                      [7, 8]]),
       size=(3, 2), nnz=2, layout=torch.sparse_coo)
>>> s[1, 0, 1]
tensor(6)
>>> s[1, 0, 1:]
tensor([6])
```

In PyTorch, the fill value of a sparse tensor cannot be specified explicitly and is assumed to be zero in general. However, there exists operations that may interpret the fill value differently. For instance, :func:`torch.sparse.softmax` computes the softmax with the assumption that the fill value is negative infinity.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 365); *backlink*

Unknown interpreted text role "func".

Sparse CSR Tensor

The CSR (Compressed Sparse Row) sparse tensor format implements the CSR format for storage of 2 dimensional tensors. Although there is no support for N-dimensional tensors, the primary advantage over the COO format is better use of storage and much faster computation operations such as sparse matrix-vector multiplication using MKL and MAGMA backends. CUDA support does not exist as of now.

A CSR sparse tensor consists of three 1-D tensors: `crow_indices`, `col_indices` and `values`:

- The `crow_indices` tensor consists of compressed row indices. This is a 1-D tensor of size `size[0] + 1`. The last element is the number of non-zeros. This tensor encodes the index in `values` and `col_indices` depending on where the given row starts. Each successive number in the tensor subtracted by the number before it denotes the number of elements in a given row.
- The `col_indices` tensor contains the column indices of each value. This is a 1-D tensor of size `nnz`.
- The `values` tensor contains the values of the CSR tensor. This is a 1-D tensor of size `nnz`.

Note

The index tensors `crow_indices` and `col_indices` should have element type either `torch.int64` (default) or `torch.int32`. If you want to use MKL-enabled matrix operations, use `torch.int32`. This is as a result of the default linking of pytorch being with MKL LP64, which uses 32 bit integer indexing.

Construction of CSR tensors

Sparse CSR matrices can be directly constructed by using the :func:`torch.sparse_csr_tensor` method. The user must supply the row and column indices and values tensors separately. The `size` argument is optional and will be deduced from the the `crow_indices` and `col_indices` if it is not present.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 407); *backlink*

Unknown interpreted text role "func".

```
>>> crow_indices = torch.tensor([0, 2, 4])
>>> col_indices = torch.tensor([0, 1, 0, 1])
>>> values = torch.tensor([1, 2, 3, 4])
>>> csr = torch.sparse_csr_tensor(crow_indices, col_indices, values, dtype=torch.double)
>>> csr
tensor(crow_indices=tensor([0, 2, 4]),
      col_indices=tensor([0, 1, 0, 1]),
      values=tensor([1., 2., 3., 4.]), size=(2, 2), nnz=4,
      dtype=torch.float64)
>>> csr.to_dense()
tensor([[1., 2.],
        [3., 4.]], dtype=torch.float64)
```

CSR Tensor Operations

The simplest way of constructing a sparse CSR tensor from a strided or sparse COO tensor is to use :meth:`tensor.to_sparse_csr`. Any zeros in the (strided) tensor will be interpreted as missing values in the sparse tensor:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 428); *backlink*

Unknown interpreted text role "meth".

```
>>> a = torch.tensor([[0, 0, 1, 0], [1, 2, 0, 0], [0, 0, 0, 0]], dtype = torch.float64)
>>> sp = a.to_sparse_csr()
>>> sp
tensor(crow_indices=tensor([0, 1, 3, 3]),
      col_indices=tensor([2, 0, 1]),
      values=tensor([1., 1., 2.]), size=(3, 4), nnz=3, dtype=torch.float64)
```

The sparse matrix-vector multiplication can be performed with the :meth:`tensor.matmul` method. This is currently the only math operation supported on CSR tensors.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 439); *backlink*

Unknown interpreted text role "meth".

```
>>> vec = torch.randn(4, 1, dtype=torch.float64)
>>> sp.matmul(vec)
tensor([[0.9078],
        [1.3180],
        [0.0000]], dtype=torch.float64)
```

Supported Linear Algebra operations

The following table summarizes supported Linear Algebra operations on sparse matrices where the operands layouts may vary. Here `T[layout]` denotes a tensor with a given layout. Similarly, `M[layout]` denotes a matrix (2-D PyTorch tensor), and `V[layout]` denotes a vector (1-D PyTorch tensor). In addition, `f` denotes a scalar (float or 0-D PyTorch tensor), `*` is element-wise multiplication, and `@` is matrix multiplication.

PyTorch operation	**Sparse grad?**	**Layout signature**

PyTorch operation	Sparse grad?	Layout signature
:func:`torch.mv` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 465); *backlink* Unknown interpreted text role "func".	no	`M[sparse_coo] @ V[strided] -> V[strided]`
:func:`torch.mv` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 465); *backlink* Unknown interpreted text role "func".	no	`M[sparse_csr] @ V[strided] -> V[strided]`
:func:`torch.matmul` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 465); *backlink* Unknown interpreted text role "func".	no	`M[sparse_coo] @ M[strided] -> M[strided]`

PyTorch operation	Sparse grad?	Layout signature
:func:`torch.matmul` **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 465); *backlink*** Unknown interpreted text role "func".	no	`M[sparse_csr] @ M[strided] -> M[strided]`
:func:`torch.mm` **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 465); *backlink*** Unknown interpreted text role "func".	no	`M[sparse_coo] @ M[strided] -> M[strided]`
:func:`torch.sparse.mm` **System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 465); *backlink*** Unknown interpreted text role "func".	yes	`M[sparse_coo] @ M[strided] -> M[strided]`

PyTorch operation	Sparse grad?	Layout signature
:func:`torch.smm` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 465); *backlink* Unknown interpreted text role "func".	no	`M[sparse_coo] @ M[strided] -> M[sparse_coo]`
:func:`torch.hspmm` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 465); *backlink* Unknown interpreted text role "func".	no	`M[sparse_coo] @ M[strided] -> M[hybrid sparse_coo]`
:func:`torch.bmm` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 465); *backlink* Unknown interpreted text role "func".	no	`T[sparse_coo] @ T[strided] -> T[strided]`

PyTorch operation	Sparse grad?	Layout signature
:func:`torch.addmm` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 465); *backlink* Unknown interpreted text role "func".	no	`f * M[strided] + f * (M[sparse_coo] @ M[strided]) -> M[strided]`
:func:`torch.sparse.addmm` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 465); *backlink* Unknown interpreted text role "func".	yes	`f * M[strided] + f * (M[sparse_coo] @ M[strided]) -> M[strided]`
:func:`torch.sspaddmm` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 465); *backlink* Unknown interpreted text role "func".	no	`f * M[sparse_coo] + f * (M[sparse_coo] @ M[strided]) -> M[sparse_coo]`

PyTorch operation	Sparse grad?	Layout signature
:func:`torch.lobpcg` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 465); *backlink* Unknown interpreted text role "func".	no	`GENEIG(M[sparse_coo]) -> M[strided], M[strided]`
:func:`torch.pca_lowrank` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 465); *backlink* Unknown interpreted text role "func".	yes	`PCA(M[sparse_coo]) -> M[strided], M[strided], M[strided]`
:func:`torch.svd_lowrank` **System Message: ERROR/3** (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 465); *backlink* Unknown interpreted text role "func".	yes	`SVD(M[sparse_coo]) -> M[strided], M[strided], M[strided]`

where "Sparse grad?" column indicates if the PyTorch operation supports backward with respect to sparse matrix argument. All PyTorch operations, except :func:`torch.smm`, support backward with respect to strided matrix arguments.

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 481); *backlink*

Unknown interpreted text role "func".

Note

Currently, PyTorch does not support matrix multiplication with the layout signature `M[strided] @ M[sparse_coo]`. However, applications can still compute this using the matrix relation `D @ S == (S.t() @`

```
D.t()).t().
```

Tensor methods and sparse

The following Tensor methods are related to sparse tensors:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 498)

Unknown directive type "autosummary".

```
.. autosummary::
    :nosignatures:

    Tensor.is_sparse
    Tensor.dense_dim
    Tensor.sparse_dim
    Tensor.sparse_mask
    Tensor.to_sparse
    Tensor.to_sparse_coo
    Tensor.to_sparse_csr
    Tensor.indices
    Tensor.values
```

The following Tensor methods are specific to sparse COO tensors:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 513)

Unknown directive type "autosummary".

```
.. autosummary::
    :toctree: generated
    :nosignatures:

    Tensor.coalesce
    Tensor.sparse_resize_
    Tensor.sparse_resize_and_clear_
    Tensor.is_coalesced
    Tensor.to_dense
```

The following methods are specific to :ref:`sparse CSR tensors <sparse-csr-docs>`:

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 523); *backlink*

Unknown interpreted text role "ref".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 525)

Unknown directive type "autosummary".

```
.. autosummary::
    :nosignatures:

    Tensor.crow_indices
    Tensor.col_indices
```

The following Tensor methods support sparse COO tensors:

:meth:`~torch.Tensor.add` :meth:`~torch.Tensor.add_` :meth:`~torch.Tensor.addmm` :meth:`~torch.Tensor.addmm_` :meth:`~torch.Tensor.any` :meth:`~torch.Tensor.asin` :meth:`~torch.Tensor.asin_` :meth:`~torch.Tensor.arcsin` :meth:`~torch.Tensor.arcsin_` :meth:`~torch.Tensor.bmm` :meth:`~torch.Tensor.clone` :meth:`~torch.Tensor.deg2rad` :meth:`~torch.Tensor.deg2rad_` :meth:`~torch.Tensor.detach` :meth:`~torch.Tensor.detach_` :meth:`~torch.Tensor.dim` :meth:`~torch.Tensor.div` :meth:`~torch.Tensor.div_` :meth:`~torch.Tensor.floor_divide` :meth:`~torch.Tensor.floor_divide_` :meth:`~torch.Tensor.get_device` :meth:`~torch.Tensor.index_select` :meth:`~torch.Tensor.isnan` :meth:`~torch.Tensor.log1p` :meth:`~torch.Tensor.log1p_` :meth:`~torch.Tensor.mm` :meth:`~torch.Tensor.mul` :meth:`~torch.Tensor.mul_` :meth:`~torch.Tensor.mv` :meth:`~torch.Tensor.narrow_copy` :meth:`~torch.Tensor.neg` :meth:`~torch.Tensor.neg_` :meth:`~torch.Tensor.negative` :meth:`~torch.Tensor.negative_` :meth:`~torch.Tensor.numel` :meth:`~torch.Tensor.rad2deg`

:meth:`~torch.Tensor.rad2deg_` :meth:`~torch.Tensor.resize_as_` :meth:`~torch.Tensor.size` :meth:`~torch.Tensor.pow` :meth:`~torch.Tensor.sqrt` :meth:`~torch.Tensor.square` :meth:`~torch.Tensor.smm` :meth:`~torch.Tensor.sspaddmm` :meth:`~torch.Tensor.sub` :meth:`~torch.Tensor.sub_` :meth:`~torch.Tensor.t` :meth:`~torch.Tensor.t_` :meth:`~torch.Tensor.transpose` :meth:`~torch.Tensor.transpose_` :meth:`~torch.Tensor.zero_`

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, **line 533**); ***backlink***

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, **line 533**); ***backlink***

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, **line 533**); ***backlink***

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, **line 533**); ***backlink***

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, **line 533**); ***backlink***

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, **line 533**); ***backlink***

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, **line 533**); ***backlink***

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, **line 533**); ***backlink***

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, **line 533**); ***backlink***

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, **line 533**); ***backlink***

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, **line 533**); ***backlink***

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, **line 533**); ***backlink***

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 533); *backlink*

Unknown interpreted text role "meth".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst, line 533); *backlink*

Unknown interpreted text role "meth".

Torch functions specific to sparse Tensors

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst, line 587)

Unknown directive type "autosummary".

```
.. autosummary::
    :toctree: generated
    :nosignatures:

    sparse_coo_tensor
    sparse_csr_tensor
    sparse.sum
    sparse.addmm
    sparse.sampled_addmm
    sparse.mm
    sspaddmm
    hspmm
    smm
    sparse.softmax
    sparse.log_softmax
```

Other functions

The following :mod:`torch` functions support sparse tensors:

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst, line 606); *backlink*

Unknown interpreted text role "mod".

:func:`~torch.cat` :func:`~torch.dstack` :func:`~torch.empty` :func:`~torch.empty_like` :func:`~torch.hstack` :func:`~torch.index_select` :func:`~torch.is_complex` :func:`~torch.is_floating_point` :func:`~torch.is_nonzero` :func:`~torch.is_same_size` :func:`~torch.is_signed` :func:`~torch.is_tensor` :func:`~torch.lobpcg` :func:`~torch.mm` :func:`~torch.native_norm` :func:`~torch.pca_lowrank` :func:`~torch.select` :func:`~torch.stack` :func:`~torch.svd_lowrank` :func:`~torch.unsqueeze` :func:`~torch.vstack` :func:`~torch.zeros` :func:`~torch.zeros_like`

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst, line 608); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst, line 608); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst, line 608); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst, line 608); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst, line 608); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 608); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 608); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 608); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 608); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 608); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 608); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 608); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 608); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 608); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 608); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 608); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 608); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 608); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 608); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 608); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 608); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 608); *backlink*

Unknown interpreted text role "func".

System Message: ERROR/3 (`D:\onboarding-resources\sample-onboarding-resources\pytorch-master\docs\source\[pytorch-master][docs][source]sparse.rst`, line 608); *backlink*

Unknown interpreted text role "func".