

pytorch-合并和切割

合并 (Merge)

cat

`cat(tensors, dim=0)` : 将多个张量合并起来, `tensor`是传入的张量, 这些张量除了在需要合并的维度上不同外, 其他维度上都必需要相同。`dim` 是指需要合并的维度。

```
In [1]: 1 import torch

In [2]: 1 a = torch.rand(4, 32, 8)

In [3]: 1 b = torch.rand(5, 32, 8)

In [4]: 1 torch.cat([a, b], dim=0).shape

torch.Size([9, 32, 8])
```

下面是一些示例:

```
In [8]: 1 a1 = torch.rand(4, 3, 32, 32)
        2 a2 = torch.rand(5, 3, 32, 32)

In [9]: 1 torch.cat([a1, a2], dim=0).shape

torch.Size([9, 3, 32, 32])
```

```
In [11]: 1 a3 = torch.rand(4, 1, 32, 32)
        2 torch.cat([a1, a3], dim=0).shape
```

```
RuntimeError                                Traceback (most recent call last)
<ipython-input-11-d47a3683e78f> in <module>
      1 a3 = torch.rand(4, 1, 32, 32)
----> 2 torch.cat([a1, a3], dim=0).shape

RuntimeError: Sizes of tensors must match except in dimension 0. Got 3 and 1 in dimension 1 (The offending index is 1)
```

```
In [12]: 1 torch.cat([a1, a3], dim=1).shape

torch.Size([4, 4, 32, 32])
```

```
In [14]: 1 a1 = torch.rand(4, 3, 16, 32)
        2 a2 = torch.rand(4, 3, 16, 32)
        3 torch.cat([a1, a2], dim = 2).shape

torch.Size([4, 3, 32, 32])
```

stack

stack(tensors, dim=0) : 不同与**cat**, 该函数会在**dim**维度上新建一个维度。同时, 需要传入的**tensor**的维度信息也一样。

适用场景: **cat**相当于同一个班级的两个老师各自统计了一部分学生, 然后合并起来。**stack**相当于两个班级的老师各自将自己班级的学生成绩统计了, 但是需要分开来合并, 表示是不同班级的。

```
In [14]: 1 a1 = torch.rand(4, 3, 16, 32)
          2 a2 = torch.rand(4, 3, 16, 32)
          3 torch.cat([a1, a2], dim = 2). shape
```

```
torch.Size([4, 3, 32, 32])
```

```
In [15]: 1 torch.stack([a1, a2], dim=2). shape
```

```
torch.Size([4, 3, 2, 16, 32])
```

```
In [17]: 1 a = torch.rand(32, 8)
          2 b = torch.rand(32, 8)
```

```
In [18]: 1 torch.stack([a, b], dim=0). shape
```

```
torch.Size([2, 32, 8])
```

切割(split)

length

split(tensor, split_size_or_sections, dim=0) :

- **tesnor: input**, 待分输入
- **split_size_or_sections**: 需要切分的大小(int or list)
- **dim**: 切分维度

当**split_size_or_sections**为**int**时, 则范围为 $(n/2, n-1)$, n 为相应维度的数字。会自动根据所给数字给划分为两个部分。

```
In [39]: 1 a = torch.rand(8, 32, 3)

In [46]: 1 x, y = a.split(3, dim=0)
        2 x.shape, y.shape
```

```
-----
ValueError                                Traceback (most recent call last)
<ipython-input-46-1daf5d87c306> in <module>
----> 1 x, y = a.split(3, dim=0)
      2 x.shape, y.shape

ValueError: too many values to unpack (expected 2)
```

```
In [47]: 1 x, y = a.split(4, dim=0)
        2 x.shape, y.shape
```

```
(torch.Size([4, 32, 3]), torch.Size([4, 32, 3]))
```

```
In [49]: 1 x, y = a.split(7, dim=0)
        2 x.shape, y.shape
```

```
(torch.Size([7, 32, 3]), torch.Size([1, 32, 3]))
```

当`split_size_or_sections`为**list**时，那么**tensor**结构会一共切分成`len(list)`这么多的小块，每个小块中的大小按照**list**中的大小决定，其中**list**中的数字总和应等于该维度的大小，否则会报错（注意这里与`split_size_or_sections`为**int**时的情况不同）。

这个**list**里面的数字只能有两个。超过两个会报错

```
In [50]: 1 x, y = a.split([1, 7], dim=0)
        2 x.shape, y.shape
```

```
(torch.Size([1, 32, 3]), torch.Size([7, 32, 3]))
```

```
In [51]: 1 x, y = a.split([5, 3], dim=0)
        2 x.shape, y.shape
```

```
(torch.Size([5, 32, 3]), torch.Size([3, 32, 3]))
```

num

`chunk(input, chunks, dim=0)`：返回的数量为指定轴的元素个数除**chunks**。如果指定轴的元素个数被**chunks**除不尽，那么最后一块的元素个数变少。

似乎只能将**chunks**设置为2，返回两个新的张量。

```
In [64]: 1 a = torch.rand(7, 32, 23)
```

```
In [65]: 1 a.shape
```

```
torch.Size([7, 32, 23])
```

```
In [69]: 1 aa, bb = a.chunk(2, dim=0)
```

```
In [67]: 1 aa.shape, bb.shape
```

```
(torch.Size([4, 32, 23]), torch.Size([3, 32, 23]))
```