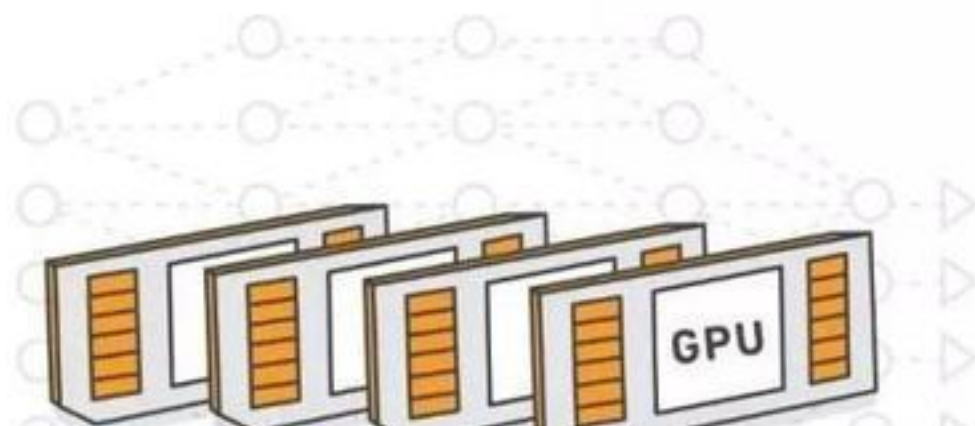


知乎

PYTORCH



## Pytorch中的torch.cat()函数



杨帆

39 人赞同了该文章

### cat() 的用法

按维数0拼接（竖着拼）

```
C = torch.cat( (A,B),0 )
```

按维数1拼接（横着拼）

```
C = torch.cat( (A,B),1 )
```

### 按维数0拼接

```
A=torch.ones(2,3)    #2x3的张量（矩阵）  
print("A:\n",A," \nA.shape:\n",A.shape," \n")
```

```
B=2*torch.ones(4,3)  #4x3的张量（矩阵）  
print("B:\n",B," \nB.shape:\n",B.shape," \n")
```

```
C=torch.cat((A,B),0)  #按维数0（行）拼接  
print("C:\n",C," \nC.shape:\n",C.shape," \n")
```

```
A:  
tensor([[1., 1., 1.],  
        [1., 1., 1.]])  
A.shape:
```

## 知乎

```
tensor([[2., 2., 2.],
        [2., 2., 2.],
        [2., 2., 2.],
        [2., 2., 2.]])
```

B.shape:  
torch.Size([4, 3])

C:  
tensor([[1., 1., 1.],
 [1., 1., 1.],
 [2., 2., 2.],
 [2., 2., 2.],
 [2., 2., 2.],
 [2., 2., 2.]])

C.shape:  
torch.Size([6, 3])

### 按维数1拼接

```
A=torch.ones(2,3)    #2x3的张量（矩阵）
print("A:\n",A,"\nA.shape:\n",A.shape,"\n")
```

```
B=2*torch.ones(2,4)  #4x3的张量（矩阵）
print("B:\n",B,"\nB.shape:\n",B.shape,"\n")
```

```
C=torch.cat((A,B),1) #按维数0（行）拼接
print("C:\n",C,"\nC.shape:\n",C.shape,"\n")
```

A:  
tensor([[1., 1., 1.],
 [1., 1., 1.]])

A.shape:  
torch.Size([2, 3])

B:  
tensor([[2., 2., 2., 2.],
 [2., 2., 2., 2.]])

B.shape:  
torch.Size([2, 4])

C:  
tensor([[1., 1., 1., 2., 2., 2., 2.],
 [1., 1., 1., 2., 2., 2., 2.]])

C.shape:  
torch.Size([2, 7])

发布于 2020-03-24 20:43

[TensorLayer \(深度学习库\)](#) [PyTorch](#) [深度学习 \(Deep Learning\)](#)