

# PyTorch的nn.Linear () 详解

原创

风雪夜归人

于 2019-11-02 17:10:06 发布

221027

收藏

版权


o

布

1007

分类专栏: 

PyTorch

 PyTorch 专栏收录该内容

7 订阅 9 篇文章

订阅专栏

PyTorch 的 `nn.Linear ()` 是用于设置网络中的全连接层的，需要注意在二维图像处理的任务中，全连接层的输入与输出一般都设置为二维张量，形状通常为 `[batch_size, size]`，不同于卷积层要求输入输出是四维张量。其用法与形参说明如下：

**CLASS** `torch.nn.Linear(in_features, out_features, bias=True)`

Applies a linear transformation to the incoming data:  $y = xA^T + b$

Parameters

- `in_features` – size of each input sample
- `out_features` – size of each output sample
- `bias` – If set to `False`, the layer will not learn an additive bias. Default: `True`

`in_features` 指的是输入的二维张量的大小，即输入的 `[batch_size, size]` 中的 `size`。

`out_features` 指的是输出的二维张量的大小，即输出的二维张量的形状为 `[batch_size, output_size]`，当然，它也代表了该全连接层的神经元个数。

从输入输出的张量的shape角度来理解，相当于一个输入为 `[batch_size, in_features]` 的张量变换成了 `[batch_size, out_features]` 的输出张量。

用法示例：

```
1 import torch as t
2 from torch import nn
3
4 # in_features由输入张量的形状决定, out_features则决定了输出张量的形状
5 connected_layer = nn.Linear(in_features = 64*64*3, out_features = 1)
6
7 # 假定输入的图像形状为[64, 64, 3]
8 input = t.randn(1, 64, 64, 3)
9
10 # 将四维张量转换为二维张量之后, 才能作为全连接层的输入
11 input = input.view(1, 64*64*3)
12 print(input.shape)
13 output = connected_layer(input) # 调用全连接层
14 print(output.shape)
```

这段代码运行结果为：

```
1 input shape is %s torch.Size([1, 12288])
2 output shape is %s torch.Size([1, 1])
```

文章知识点与官方知识档案匹配，可进一步学习相关知识

Python入门技能树 人工智能 深度学习 32786 人正在系统学习中

 创作挑战赛

新人创作奖励来咯，坚持创作打卡瓜分现金大奖

>

 风雪夜归人o

关注

 324