在 PyTorch 中，Linear 类是一个全连接层（也称为线性层或密集层），用于实现线性变换。它将输入数据通过一个线性函数（即矩阵乘法加上偏置）映射到输出空间。

作用

Linear 层的作用是对输入数据进行线性变换，公式如下：

[\text{output} = \text{input} \times \text{weight}^\top + \text{bias}]

其中：

- input 是输入数据。
- weight 是权重矩阵。
- bias 是偏置向量。

用法

Linear 类位于 torch.nn 模块中，使用时需要指定输入特征的数量和输出特征的数量。

1. 导入 Linear 类

```
import torch
import torch.nn as nn
```

2. 创建 Linear 层

```
# 创建一个线性层，输入特征数为 in_features，输出特征数为 out_features
linear_layer = nn.Linear(in_features, out_features)
```

- in_features：输入数据的特征数量。
- out_features：输出数据的特征数量。

3. 使用 Linear 层

```
# 假设输入数据是一个大小为 (batch_size, in_features) 的张量
input_data = torch.randn(batch_size, in_features)

# 通过线性层进行前向传播
output = linear_layer(input_data)
```

- input_data 是一个大小为 (batch_size, in_features) 的张量，其中 batch_size 是批量大小。
- output 是一个大小为 (batch_size, out_features) 的张量。

4. 查看权重和偏置

```
# 查看权重矩阵
print(linear_layer.weight)

# 查看偏置向量
print(linear_layer.bias)

```

- weight 是一个大小为 (out_features, in_features) 的可学习参数。
- bias 是一个大小为 (out_features,) 的可学习参数。

示例

以下是一个完整的示例，展示如何使用 Linear 层：

```python
import torch
import torch.nn as nn

# 定义输入特征数和输出特征数
in_features = 5
out_features = 3

# 创建线性层
linear_layer = nn.Linear(in_features, out_features)

# 创建一个大小为 (batch_size, in_features) 的输入张量
batch_size = 2
input_data = torch.randn(batch_size, in_features)

# 前向传播
output = linear_layer(input_data)

print("Input data:")
print(input_data)
print("\nOutput data:")
print(output)
print("\nWeight matrix:")
print(linear_layer.weight)
print("\nBias vector:")
print(linear_layer.bias)

```

总结

- Linear 类用于实现全连接层，对输入数据进行线性变换。
- 需要指定输入特征数 in_features 和输出特征数 out_features。
- 通过 forward 方法进行前向传播，输出数据的大小为 (batch_size, out_features)。
- 权重和偏置是可学习的参数，可以通过 linear_layer.weight 和 linear_layer.bias 访问。