

```
1 # 导入 PyTorch 库
2 import torch
3 # 从 torch 中导入 nn 模块, 用于构建神经网络
4 from torch import nn
5 # 从 torch.nn 中导入 functional 模块, 包含常用的函数
6 from torch.nn import functional as F
7
8 # 使用 nn.Sequential 构建一个顺序容器, 按顺序包含多个层
9 net = nn.Sequential(
10     # 第一个全连接层, 输入维度为 20, 输出维度为 256
11     # 当输入 X (形状为 (3, 20)) 传入该层时, nn.Linear 会对每个样本 (共 3 个样本) 进行
    线性变换。具体来说, 它会执行矩阵乘法操作, 将输入的每个样本 (形状为 (20,)) 与权重矩阵 (形状为
    (20, 256)) 相乘, 再加上偏置向量 (形状为 (256,)) , 得到输出的每个样本 (形状为 (256,)) )。
12     # 因此, 经过该层后, 输出的张量形状为 (3, 256) , 即 3 个样本, 每个样本有 256 个特征。
13     nn.Linear(20, 256),
14     # ReLU 激活函数层, 用于引入非线性
15     nn.ReLU(),
16     # 第二个全连接层, 输入维度为 256, 输出维度为 10
17     nn.Linear(256, 10)
18 )
19
20 # 生成一个形状为 (2, 20) 的随机张量 X, 模拟输入数据
21 # 主要是这里的2, 不好理解, 可以理解成有2个样本, 每个样本20个特征
22 X = torch.rand(2, 20)
23 # 将输入数据 X 传入神经网络 net 进行前向传播, 得到输出结果
24 res = net(X)
25 # 打印输出结果
26 print(res)
```