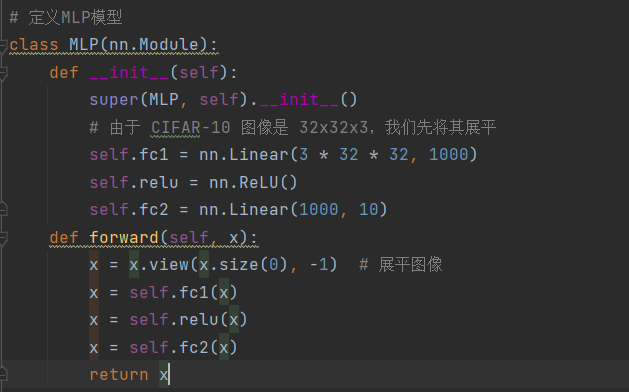
数据库选择

由于要求尽量不使用MNIST或fasion-MNIST数据库，我选择了一个替代的数据库——CIFAR-10。CIFAR-10是一个包含60,000个32x32彩色图像的数据集，分为10个类别，每个类别有6,000个图像。这个数据集因其图像多样性和复杂性而被广泛用于机器学习实验。

数据预处理和划分

我们从CIFAR-10数据集中随机选择了训练集和测试集。训练集用于模型训练，而测试集则用于评估模型的性能。我使用PyTorch的torchvision.transforms模块对图像进行必要的预处理，如归一化、数据增强等。

模型构建



这段代码定义了一个名为MLP（多层感知机）的PyTorch神经网络模型，它采用了全连接层（fully connected layers）的结构。具体来说，这个模型由以下部分组成：

**输入层**：由于CIFAR-10数据集中的图像是32x32像素的彩色图像（即每个图像有3个颜色通道：红、绿、蓝），因此每个图像有3 \* 32 \* 32个像素值。在模型中，这些像素值首先被展平（flatten）成一个一维向量，作为全连接网络的输入。

**第一个全连接层（fc1）**：self.fc1 = nn.Linear(3 \* 32 \* 32, 1000)定义了一个从输入层到隐藏层的全连接层。输入层有3 \* 32 \* 32个神经元（对应于展平后的图像像素），输出层有1000个神经元。这意味着该层有3 \* 32 \* 32 \* 1000个权重参数（加上1000个偏置参数）。

**激活函数**：self.relu = nn.ReLU()定义了一个ReLU（Rectified Linear Unit）激活函数，它用于在第一个全连接层之后添加非线性。ReLU函数将输入值中小于0的部分置为0，大于等于0的部分保持不变。

**第二个全连接层（fc2）**：self.fc2 = nn.Linear(1000, 10)定义了一个从隐藏层到输出层的全连接层。输入层有1000个神经元（对应于第一个全连接层的输出），输出层有10个神经元（对应于CIFAR-10数据集的10个类别）。因此，这一层有1000 \* 10个权重参数（加上10个偏置参数）。

**输出层**：模型的输出层是第二个全连接层的输出，它包含了10个神经元，每个神经元对应一个类别的得分或概率（虽然这些值在模型输出时通常是未归一化的，需要通过softmax函数转换为概率分布）。

总结来说，这个MLP模型是一个简单的全连接神经网络，具有一个输入层、一个隐藏层（通过ReLU激活）和一个输出层。它用于对CIFAR-10数据集中的图像进行分类。

运行结果：

