实验涉及到MNIST手写数字识别,作为一个经典的问题，MNIST数据集包含了一系列28x28像素的手写数字图像，我们的目标是使用深度学习模型来准确识别这些数字。

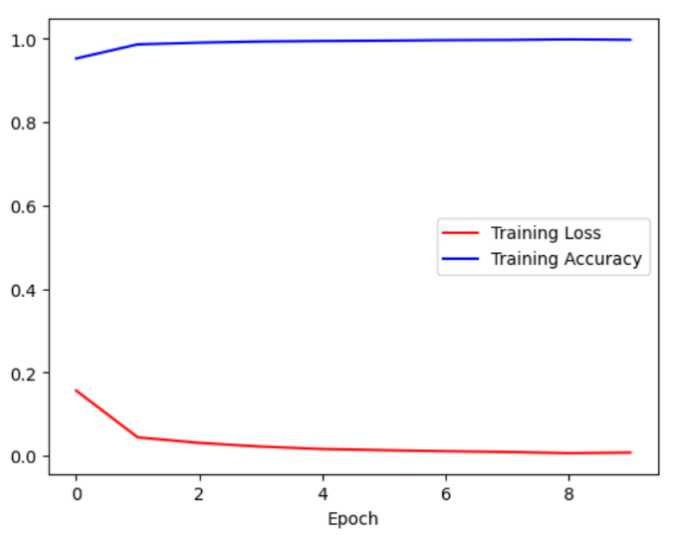
首先，我们对MNIST数据进行了预处理。通过使用transforms.Compose，我们将图像转换成张量，并进行了标准化，以确保它们在训练模型前是可用的。

**卷积神经网络（CNN）模型：**

我们设计了一个简单的卷积神经网络（CNN）模型，该模型包括卷积层、池化层和全连接层。这个结构旨在从图像中提取特征，适应手写数字的特性。

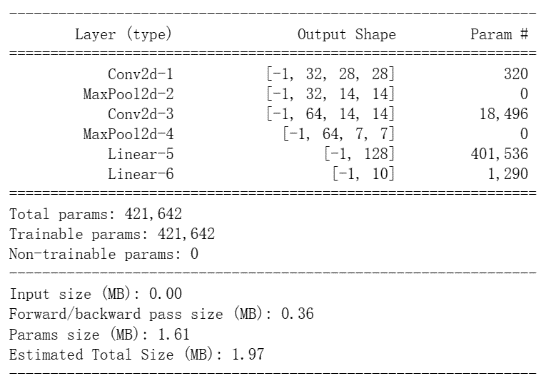
CNN训练和学习曲线：

我们使用Adam优化器和交叉熵损失函数对CNN模型进行了训练。通过迭代训练数据，我们记录了每个epoch的损失和准确度，并使用matplotlib库绘制了学习曲线，如下图所示：



CNN模型结构总结：

通过使用torchsummary库，我们打印了CNN模型的结构和参数数量。这有助于我们了解模型的规模和复杂性。



首先，我们有一个卷积层（Conv2d），它有32个卷积核，每个卷积核的大小为3x3。这一层负责提取输入图像的低级特征，如边缘和纹理。我们有320个可学习的参数，用于调整卷积核的权重。

接着，通过最大池化层（MaxPool2d），我们将卷积层的输出尺寸减小一半，使得模型能够更好地处理大规模图像而不牺牲计算效率。

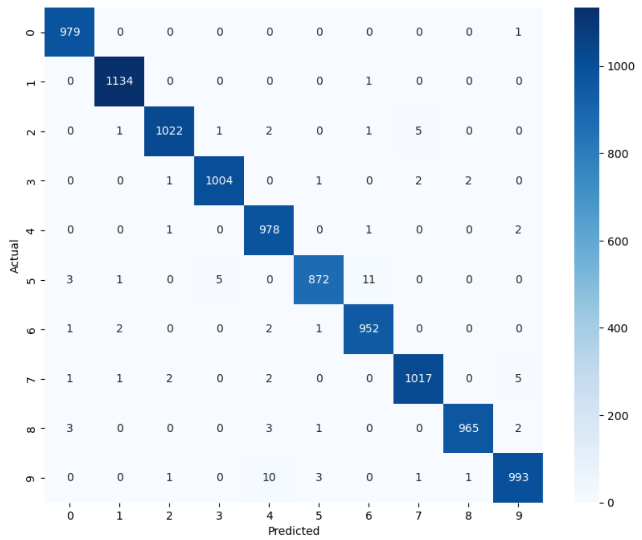
然后，我们再次使用卷积层，但这次我们增加了卷积核的数量，有64个。同样，这一层之后还有一个最大池化层。通过这一系列操作，我们使网络能够逐渐捕捉到更抽象的特征。

接下来，我们有两个全连接层（Linear），分别包含128和10个神经元。这些层负责将卷积层提取的特征映射到我们感兴趣的10个输出类别。总共有401,536个参数用于调整这些全连接层的权重。

整个模型总共有421,642个可训练的参数，而我们使用了Adam优化器来在训练过程中调整这些参数，使得网络能够学到更准确的特征，从而实现对手写数字的准确分类。

CNN混淆矩阵和热力图分析：

在测试集上进行了预测，生成了混淆矩阵，并使用seaborn绘制了热力图。混淆矩阵分析帮助我们深入了解模型在每个类别上的性能。

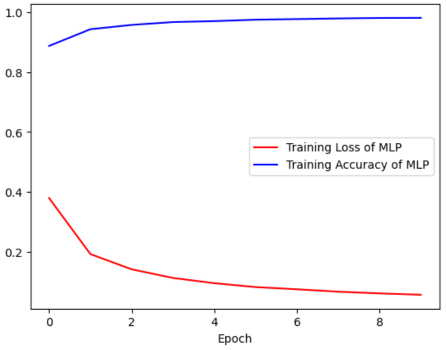


**全连接神经网络（MLP）模型：**

为了对比，我们还设计了一个简单的全连接神经网络（MLP）模型，用于进一步验证我们的实验结果。

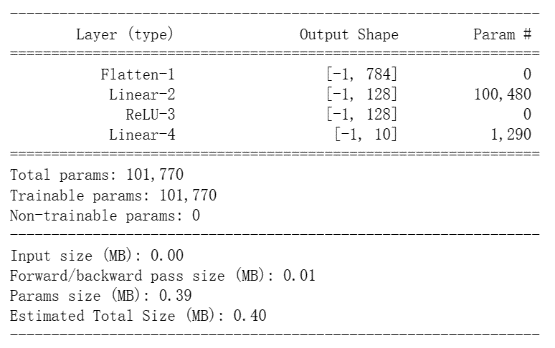
MLP训练和学习曲线：

对MLP模型进行了相似的训练过程，并绘制了相应的学习曲线。



MLP模型结构总结：

通过使用torchsummary库，我们打印了MLP模型的结构和参数数量。



首先，让我们看一下模型的层次结构。我们的输入是28x28像素的图像，首先通过一个Flatten层，将图像展平成一个784维的向量。接下来，我们有一个全连接层，它将这个784维的向量映射到128个特征。这个过程中，我们有约100,480个可学习的参数，负责调整模型以更好地拟合训练数据。

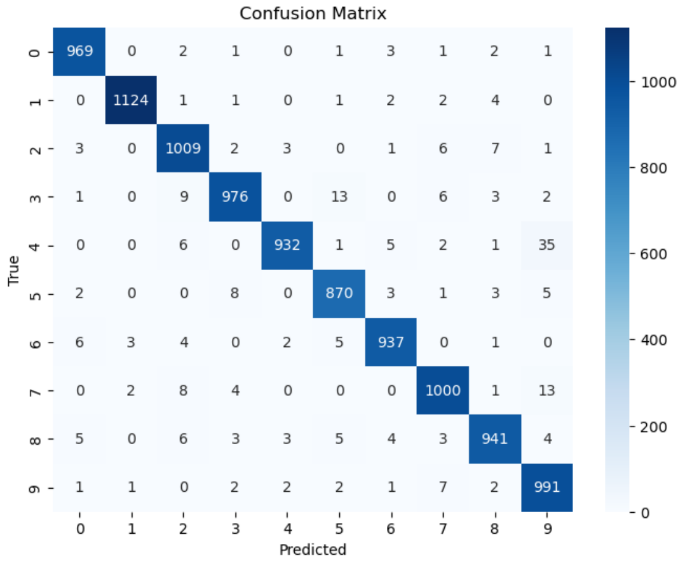
在这之后，我们使用ReLU激活函数对输出进行非线性变换，然后再连接到最终的全连接层。这个层将128个特征映射到我们感兴趣的10个输出，对应于MNIST数据集的10个数字类别。

最终，这个模型总共有101,770个可训练的参数。在训练的过程中，我们使用了Adam优化器来调整这些参数，以便我们的模型更好地适应手写数字图像的特征。

总的来说，这个结构是相当直观的，因为我们从简单的图像表示开始，逐渐提取出更高级别的特征，最终使模型能够正确地分类手写数字。

MLP混淆矩阵和热力图分析：

同样，在MLP模型上进行了混淆矩阵分析和热力图绘制，以进一步对比两个模型的性能。



**总结**：

综上所述，我们的实验结果表明，CNN模型在MNIST手写数字识别任务上表现出色。其准确度和学习曲线都优于MLP模型。CNN通过利用图像的空间信息和局部特征，更好地适应了问题的复杂性。