缪亚泰 计算机1602 1611640213

卷积神经网络实现量子态图像分类

1. **问题背景重述**

**1.1问题的背景**

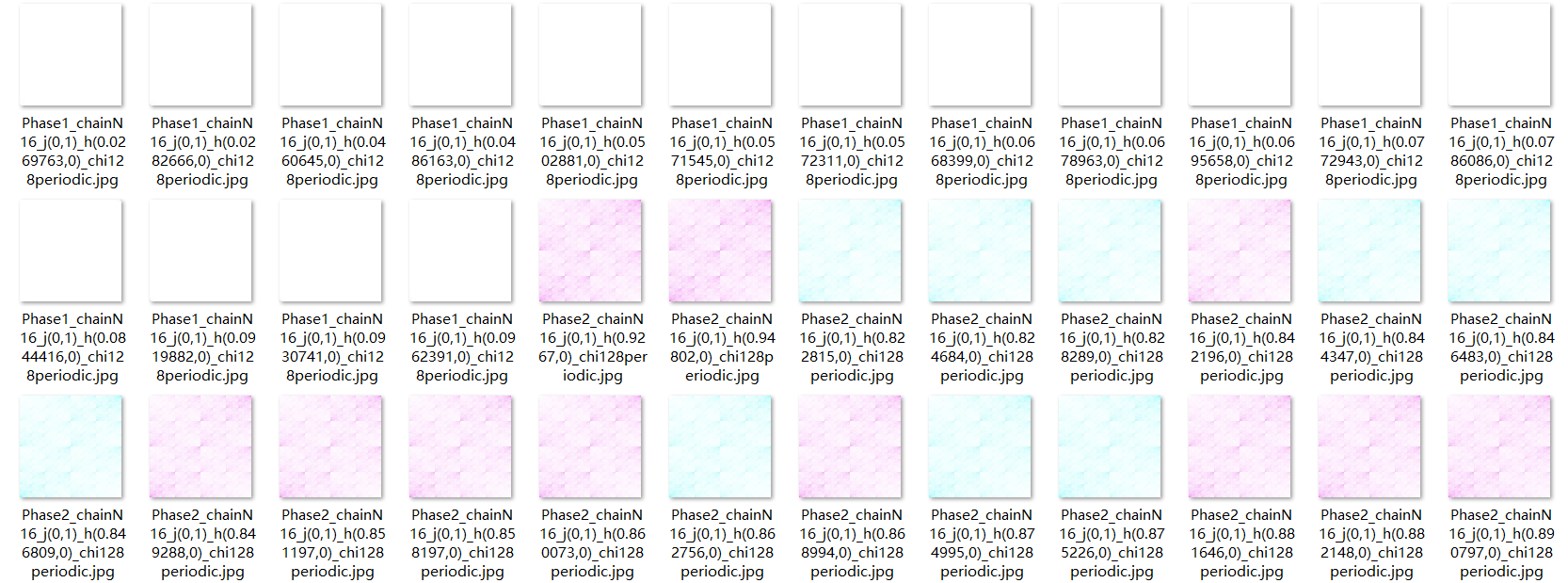
近些年来，图像分类在模式识别领域占据着重要的地位。迅速增长的图像数据对于图像信息的分析与处理提出了新的要求。卷积神经网络应运而生，以其强大的图像识别分类能力被广泛的应用于各种图像分类系统，并取得了十分显著的效果。

**1.2问题的重述**

使用Pytorch搭建深度卷积神经网络，利用Python语言编程实现卷积神经网络模型，利用量子态图像数据集中给出的训练集训练模型，并在测试集进行分类已验证模型的效能。

1. **数据集的描述**

本次实验中所使用的数据集为自旋为0.5的量子态图片图像数据集，此数据集共有100张彩色图像，这些图像尺寸为3\*256\*256，即尺寸为256×256像素的3通道rgb彩色图像。这些图片共分为2个类，分别为：‘Phase1’，‘Phase2’。测试集80张，每类有40张图。训练集20张，每类有10张图。训练集划分为10个批次，每个批次8张图；测试集划分为两个批次，每个批次10张图。并且这些类完全互相排斥，任意两类之间没有重叠，即‘Phase1’，‘Phase2’完全不同。图1为自旋为0.5量子态图像实例。



**图1：自旋为0.5量子态图像数据集**

1. **模型的建立与求解**

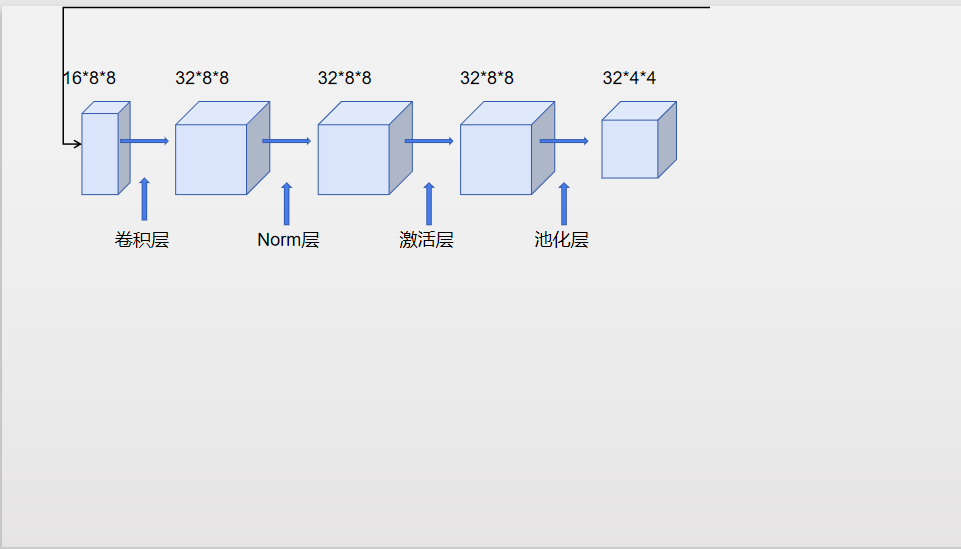
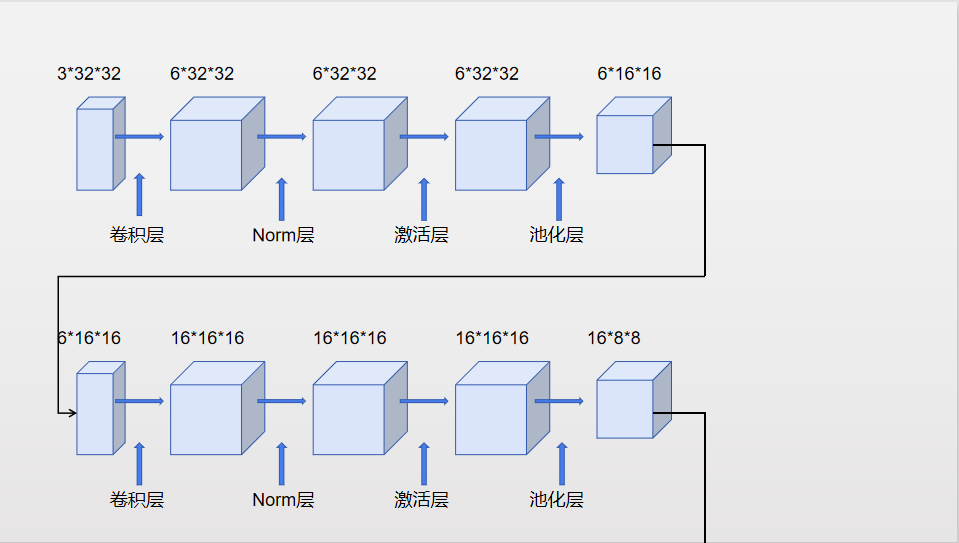
**3.1模型的建立**

**Pytorch卷积模块介绍：**

Pytorch作为一个深度学习库，卷积神经网络是其中最基础的一个模块，卷积神经网络中所有层结构都可以通过nn这个包调用。

**模型的搭建：**

本人利用Pytorch卷积模块搭建了一个六层网络，分为三层卷积层和三层全连接层，每层卷积网络分为四个部分，分别为卷积层，Norm层（归一化层），激活层，池化层。最后加上三层全连接层。

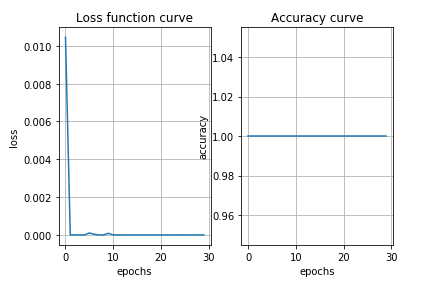


**图2：卷积网络架构图**

**3.2模型的求解：**

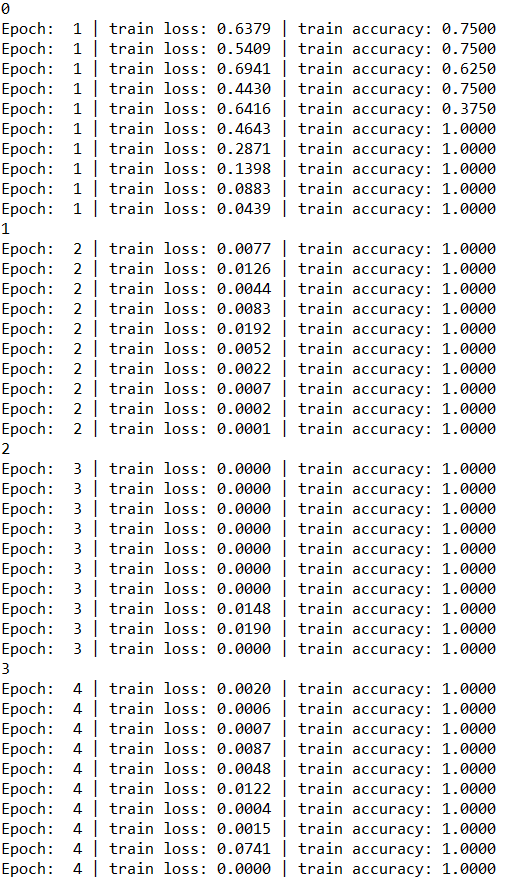
**3.2.1绘制损失函数值及准确率变化曲线图：**

训练阶段每轮每次选取8张图片进行训练，共10次完成一轮，记录本轮最后一次的损失函数值及训练准确率；共进行30轮训练，画出损失函数值及准确率变化曲线图。



**图3：损失函数值及训练准确率曲线图**

由于我选取每轮训练的最后一次损失值和准确率进行作图，而且损失值下降较快，准确率上升的也较快，使得此曲线图效果并不能较明显的看出实际效果，无法有力的说明模型训练的准确性。为此我截取了一部分程序运行时损失值和准确率的实时更新值，即为图4所展示的效果。



**图4：损失函数值及准确率程序运行图**

从图4可以看出，损失值正常下降，准确率正常上升，最初几轮训练中，损失值偶尔会有较小幅度的突升，但处于正常范围内，且整体依然呈现下降趋势，当训练继续进行时，损失值为逐渐降为0。

**3.2.2测试准确率：**

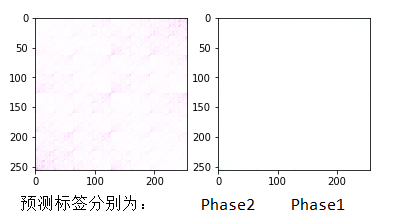
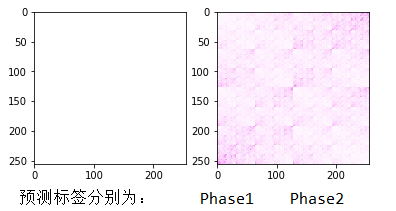
进行测试时，将测试集划分批次，每批次10张图片，即每次从20张图片中随机抽取10张进行测试，若测试结果与所给标签相符，记录，最后统计准确率。进行多次测试，最后取平均值。本次测试所得准确率平均值为92%。经过多组试验，获得多次测试准确率最后平均值大约趋近于90%。结果较好。



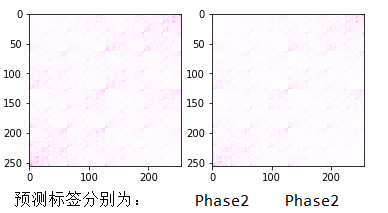
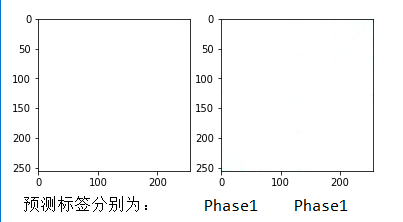
**图5：测试准确率结果图**

**3.2.3图像分类：**

预测时，将测试集划分批次，每批次2张图片，即每次从20张图片中随机抽取2张进行分类预测，以下分别为测试时出现的情况，预测结果较为准确。



**图6：预测图1 图7：预测图2**



**图8：预测图3 图9：预测图4**

由图6可知，本次测试所抽取的2张图片分别为 Phase1，Phase2。根据图6所得结果可得，预测分类正确，测试成功。图7、图8、图9分别为其他预测情况，预测分类均正确，此处仅举以上4例进行说明。

1. **模型的评价与推广**

本次实验我搭建的网络结构是一个六层经典卷积网络，模型结构较为简单，同时可以有效进行训练，测试准确率维持在90%左右。是一个较为良好的模型。

卷积神经网络模型有着较为广泛的应用空间，经过一些网络结构的调整和参数的调整后可广泛应用于其他图像的识别。

1. **总结**

通过上一次卷积网络实现Cifar10图像分类的经验，此次实验进一步加深，进行量子态图像分类任务。由于对数据集十分陌生，而且由于科学任务的严谨性，图像不能进行较多的处理，例如图像增强，图像切割等等。所以最初在加载数据集时遇到了很多麻烦。其中最大的问题是，对于数据集的陌生导致加载的数据存在各种各样的格式转换问题。同时通过这次实验我也发现了上一次实验中的很多不足之处，例如由于Pytorch存在现成的Cifar10数据集的加载导入库函数，简化了我们的编程操作，十分便利。但有利也有弊，当用新的陌生的图像数据时，会存在各种各样的格式问题。这是由于对数据加载过程的不熟练所导致的。因此本次实验中，我自己着手一步步调格式转换，最终完成数据集的加载。网络结构的搭建有了上一次实验的经验，此次过程中，并不复杂。初次运行时效果并不理想。测试准确率仅仅只有50%，此等效果下可以证明数据集没有在网络结构中进行有效的训练，使得无法对测试集进行识别分类。后来我用了一些小trick，使得测试准确率维持在90%左右。结果较为满意。但还是存在一些不足，没能达到100%的准确率，这对于科学研究是不可取的，因此我会继续改进，争取达到100%准确率。