# 实验2：使用卷积神经网络进行图像识别报告

# 实验背景

1.1实验数据

4\*4黑白像素点图片，20,000个训练数据，10,000个测试数据

1.2实验环境

Windows 11，PyCharm，python 3.10

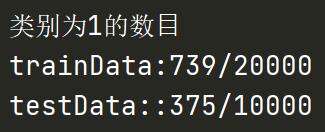
1.3实验内容

使⽤Pytorch实现⼀个卷积神经⽹络，完成⼀个图像分类任务。在4\*4的图片中，比较外围⿊⾊像素点和内圈⿊⾊像素点个数的⼤⼩，将图片分类。

1.4数据分析

此次实验使用的数据集包含20,000个训练数据和10,000个测试数据，内容为4\*4黑白像素点图片，若图片外围⿊⾊像素点个数⼤于内圈⿊⾊像素点个数，则分为0类，反之分为1类。

训练集中包含739张1类图片，测试集中包含375张1类图片，两数据集格式正常，均无缺失值。



1.5卷积神经网络图像分类可行性分析

局部感知和权值共享：CNN通过局部感知野来提取图像的局部特征，这种局部感知的机制使得网络能够专注于图像的局部信息，而局部信息对于图像分类来说是非常重要的。权值共享机制减少了模型的参数数量，使得网络更容易训练，同时也增强了模型的泛化能力。

CNN通常包含多个卷积层、池化层和全连接层，这种多层次的结构使得网络能够学习到从低级到高级的图像特征。低层次的卷积层可以捕捉到边缘、角点等低级特征，而高层次的卷积层则能够捕捉到更复杂的图像结构和模式。

由于卷积和池化操作的特性，CNN对于图像的平移、缩放和旋转具有一定的不变性，这使得CNN在处理具有不同变换的图像时有较高鲁棒性。

CNN通过反向传播算法和梯度下降等优化技术，能够自动学习到对图像分类任务有用的特征表示。这种端到端的特征学习方式避免了传统图像分类方法中繁琐的特征工程步骤。

# 实验方案设计

2.1数据集预处理

使用torchvision和PIL库处理数据集。定义一个MyDataset类，继承自torch.utils.data.Dataset，读取CSV文件，并将图像路径和标签存储到imgs列表中，同时设置transform。在\_\_getitem\_\_方法中，使用PIL打开图像，并转换为二值图像（黑白），然后应用transform。

创建训练和测试数据集实例，使用transforms.ToTensor函数将PIL图像转换为Tensor，并使用DataLoader来加载数据。在使用DataLoader时，设置shuffle=True用于训练数据，这样可以在每个epoch开始时打乱数据，有助于模型泛化。

通过对数据集进行分析可以发现，数据集中0与1类图片数量相差很大，因此定义chooseData函数对1类进⾏上采样，将原训练集进行切片，使之长度为1类图片数量的scale倍，以平衡训练集中0与1类的数量关系。

2.2构建卷积神经网络

定义一个Net类，继承自troch.nn.Module，并对\_\_init\_\_和forward方法进行重写。定义卷积层序列，设置1个卷积层，其输入输出通道数均为1，卷积核大小及步长也为1，且带有偏置项；激活函数选择ReLU，引入非线性因素，提高模型表达能力；全连接层采用输入特征数16(4\*4)，输出特征数2的线性层。

在前向传播过程中，输入量X依次经过卷积层、多维tensor展平为一维、激活函数、线性层，得到最终结果。

2.3超参数选择

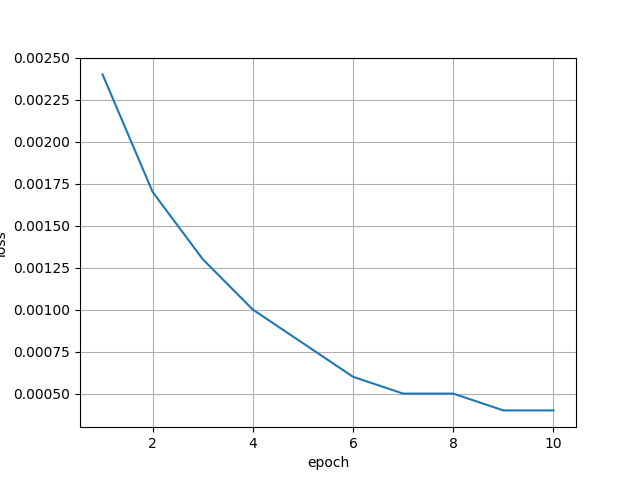
训练数据为黑白像素点图片，故输入通道数设置为1；数据图片较为简单，数据集规模较小，使用1个卷积核即可，即输出通道数设置为1；图片尺寸为4\*4，特征分类明显，而较小的卷积核更适合捕捉局部特征，适合该二分类任务，故卷积核大小为1；图片本身特征较少，故步长设置为1，否则可能丢失重要特征和细节信息。输出通道输和卷积核大小过大也会增加网络的计算成本和模型的复杂度，可能导致过拟合。

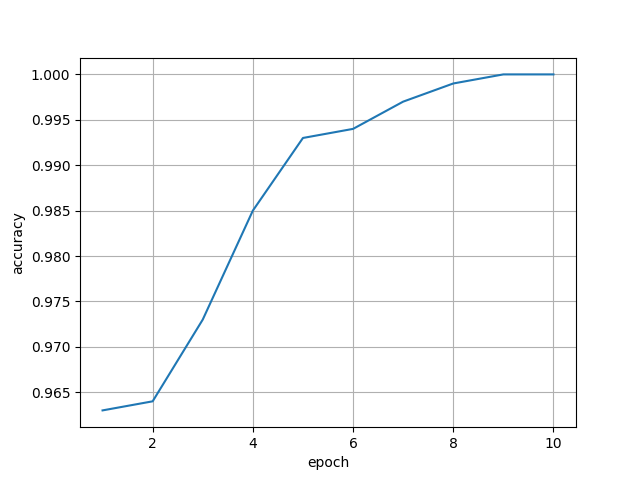
优化器选择随机梯度下降法，学习率lr从较小值（如0.01）开始，根据训练结果和运行效率进行调整，最终选择0.1作为学习率。

批处理大小（Batch Size）一般为2的幂，如32、64、128等，较大的batch\_size可以加快训练速度，但可能会导致模型精度下降；而较小的batch\_size可以提高模型精度，但会减慢训练速度。通过测试发现batch\_size为64时准确率较高。

# 实验过程

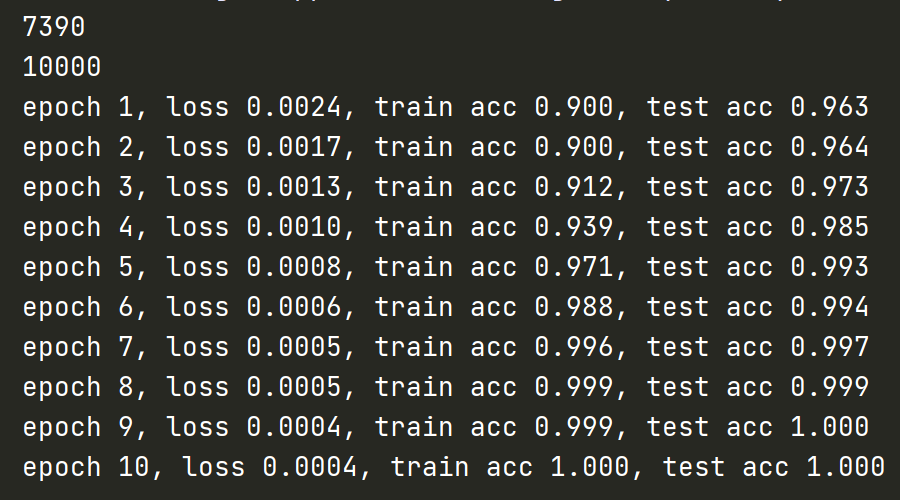
根据训练过程中输出信息可以发现，随着训练迭代次数增加，loss值在前4代完整训练中迅速降低，在epoch超过4后缓慢降低；测试准确率迅速升高，在大约5代完整训练后已经达到99%以上，之后缓慢上升并保持稳定。





# 性能测试

训练集20,000张图片，测试集10,000张图片，训练后模型在测试集上的准确率为100.0%。



# 实验习题

1. 写出两种以上不同的卷积padding策略，并说明在实验中选取的是哪一种

（1）padding策略

在卷积神经网络中，padding指的是在进行卷积操作时，对输入进行边缘填充，以控制输出特征图的大小和形状。

常见的padding策略有：

**·**无填充(Valid padding)：不填充；

**·**零填充(Zero padding) ：对图像或者张量的边缘进行补零填充操作，简单高效，但可能会导致数据信息的丢失；

**·**常数填充(Constant padding)：定义一个常数来对图像或者张量的边缘进行填充，若该常数等于0则等价于零填充，保留了数据信息，但可能导致数据分布的改变；

**·**镜像填充(Reflection padding)：对图像或者张量的边缘进行镜像对称的填充，保留了数据分布和数据信息，但计算量相对较大；

**·**重复填充(Replication padding)：重复图像的边缘像素值，将新的边界像素值用边缘像素值扩展，保留了数据分布，但可能导致数据信息的重复。

（2）在实验中，选取的padding策略为Valid padding，即无填充。