**人**

**工**

**智**

**能**

**概**

**论**

**实**

**验**

**报**

**告**

班级：2011104

学号：2201110424

姓名：李彦廷

**《人工智能概论》实验内容与要求**

**实验1 基于A\*算法的8数码问题求解**

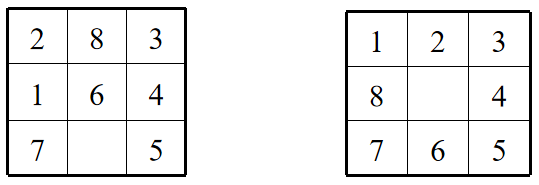
问题描述：在3\*3的棋盘中有8个数码（数字块）和一个空格，只有与空格相邻的数码能移动到空格位置。从初始状态以最小的步长移动到目标状态。

实验要求：参考A\*算法核心代码，以8数码问题为例实现A\*算法的求解程序。

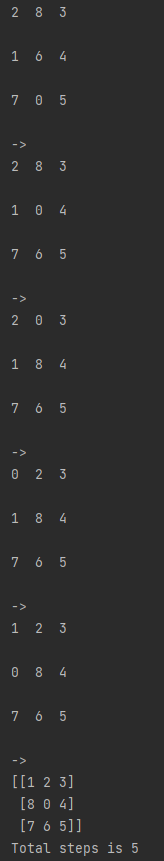
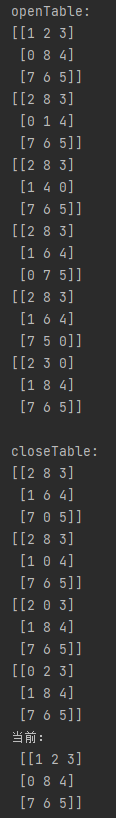
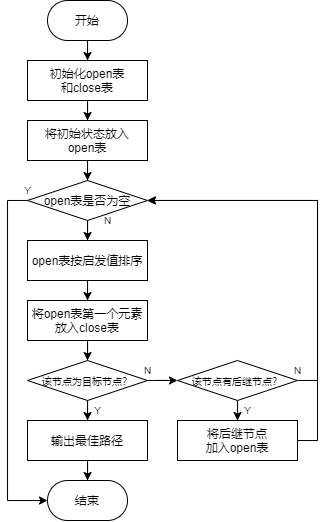
实验内容：

1. 基于参考代码，运行程序，并画出程序的总体流程图。自动统计并输出到达目标的总步数，自动记录并输出每次移动后open表与closed表的变化。

要求初始状态与目标状态分别为：



**【撰写内容】**



1. 设计两种新的启发式函数，并运行程序。对包含示例在内的3种启发式函数的特点与结果进行对比分析。

**【撰写内容】**

# 错位数码

def getFunctionValue(self, depth):

cur\_node = self.state.copy()

fin\_node = self.answer.copy()

dist = 0

N = len(cur\_node)

for i in range(N):

for j in range(N):

if cur\_node[i][j] != fin\_node[i][j]:

dist += 1

return dist + depth

# 逆转数码

def getFunctionValue(self, depth):

cur\_node = self.state.copy()

fin\_node = self.answer.copy()

dist = 0

N = len(cur\_node)

for i in range(N):

for j in range(N):

if cur\_node[i][j] != fin\_node[i][j]:

index = np.argwhere(fin\_node == cur\_node[i][j])

x = index[0][0] # 最终x距离

y = index[0][1] # 最终y距离

if cur\_node[x][y] == fin\_node[i][j]:

dist += 0.5

return 3 \* dist + depth

曼哈顿距离：可以提供相对准确的搜索指引，速度较快；但可能会受到真实路径的影响，曼哈顿距离可能会低估两点的实际距离。

错位数码：计算简单，速度快；但结果并不精确，可能导致搜索效率低。

逆转数码：可以提供相对精确的搜索指引，逆转数越多则表示与目标状态越远；逆转数的计算相对复杂，需要耗费时间。

**实验2 基于遗传算法的TSP问题求解**

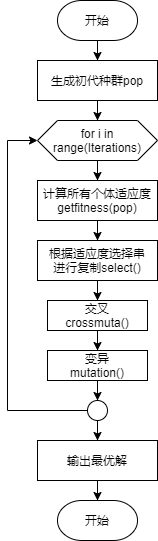
问题描述：假设有一个旅行商人要拜访N个城市，他必须选择所要走的路径，路径的限制是每个城市只能拜访一次，而且最后要回到原来出发的城市。路径的选择目标是要求得的路径路程为所有路径之中的最小值。

实验要求：参考遗传算法核心代码，以TSP问题为例实现遗传算法的求解程序。

实验内容：

1. 基于参考代码，运行程序，并画出程序的总体流程图。给出适应度函数的定义。城市数量设定为10个。

**【撰写内容】**

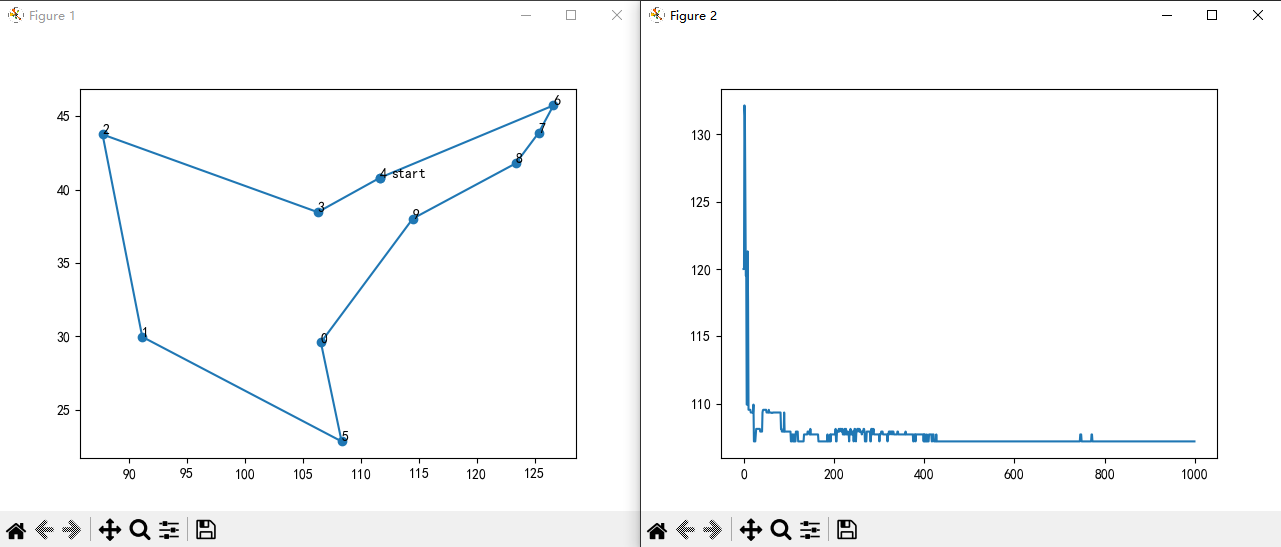


适应度函数：遗传算法中需要评定各个个体的优劣程度，从而决定其遗传机会的大小，用于这个衡量这一程度的函数即为适应度函数。

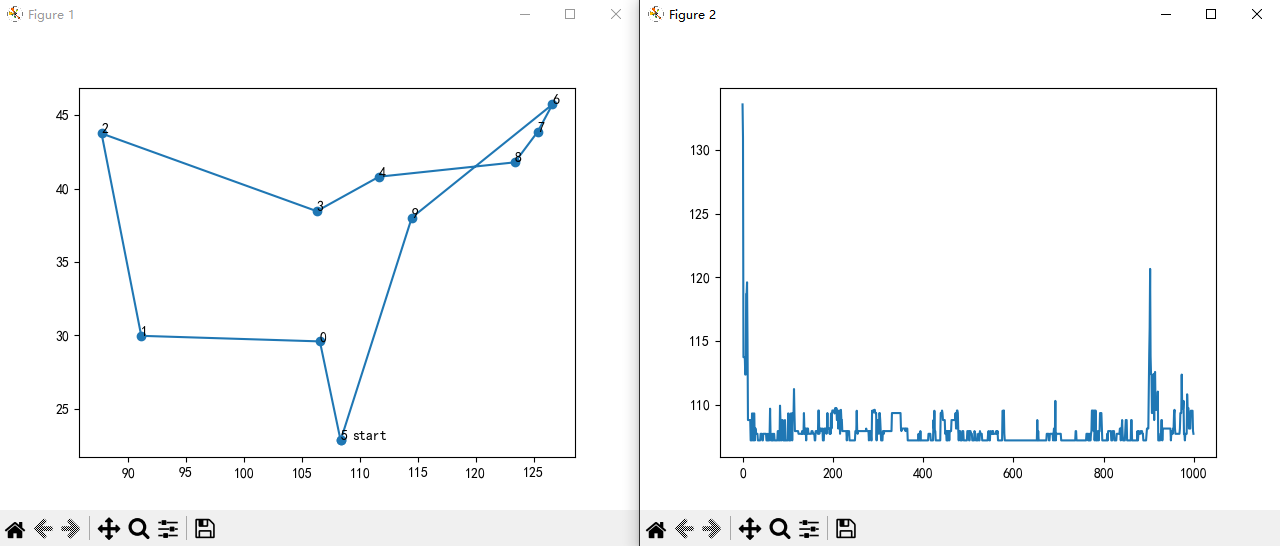
1. 基于某一个固定数量（10个）与位置的城市集合，修改种群数量、交叉概率、变异概率、迭代次数等参数，对比分析算法的特性与结果。

**【撰写内容】**

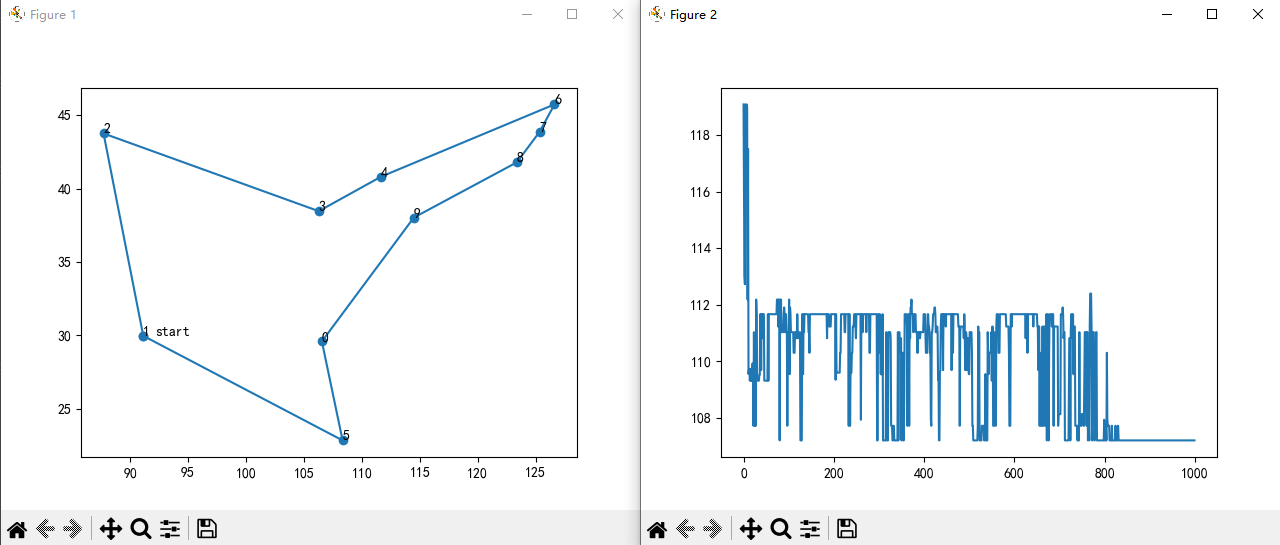
原始数据：种群大小200 交叉率0.4 变异率0.2 迭代次数1000：



变异率提高到0.8：



交叉率提高到0.9：



结论：

①群体规模太小，遗传算法的优化性能不太好，易陷入局部最优解；群体规模太大，结果越精确，但计算复杂，效率低。

②变异率太高，高适应度基因容易被破坏，造成搜索停滞收敛慢；变异率太低，可能导致算法陷入局部最优解。

③交叉率太高，会因过多的搜索不必要的解空间而耗费时间；交叉率太低，种群多样性太低，个体相似度高，算法可能过早收敛，无法继续探索解空间。

**实验3 基于神经网络的图像分类**

问题描述：基于特定图像数据集，研究人工神经网络分类器的设计与实现方法。

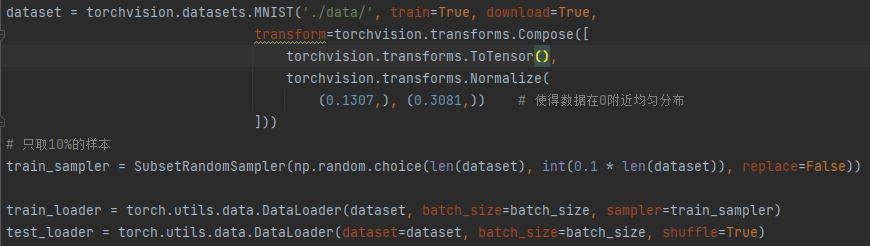
实验要求：基于MNIST手写数字图像数据集，完成神经网络模型的网络设计及参数选择，并在训练集上进行训练，在测试集上进行性能评估。了解Pytorch或TensorFlow等深度学习框架，基于上述算法库完成实验内容。

实验内容：

1. 训练集分组：从原始训练集中分别抽取10%，30%，50%，70%，100%的样本构成新的训练集，每个类别样本数量相同。可根据需要选择是否从新的训练集中抽取一定比例样本作为验证集。

**【撰写内容】**

抽取样本的关键代码如下：（选取10%样本为例）



1. 建立全连接神经网络模型，隐含层数量不超过2层。给出网络结构的所有参数设置情况，包括网络层数、每层神经元数量、活化函数的选择、学习率的设定、损失函数的定义等。对网络的输入进行描述，即原始特征或其他特征的表示方式。

**【撰写内容】**

模型由三层线性层组成（卷积见后）；

第一层为784个神经元，第二层为256个，第三层为64个，输出为10个神经元

活化函数为relu函数；

学习率设定为0.01；

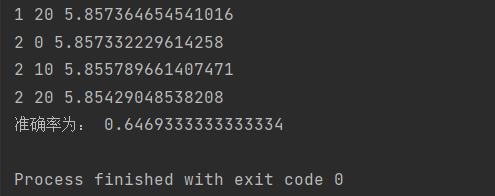
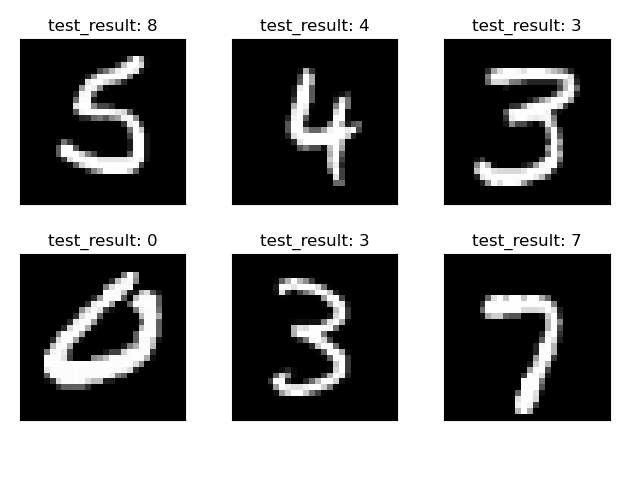
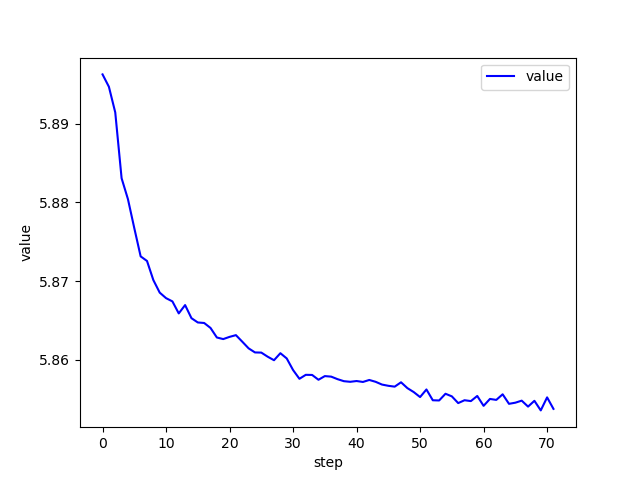
损失函数用于将模型的预测结果和真实标签对比，并输出一个表示预测误差的数值。

原始特征：网络输入为28\*28的灰度图像，并将图像转换为张量，归一到均值为01307，标准差为0.3081的分布上。

1. 基于每个训练集，训练模型，记录训练集的准确率，以图表等方式进行呈现。进一步，在完整的测试集上进行性能评测，结果以图表等方式进行呈现。对实验结果进行一定的描述与分析。

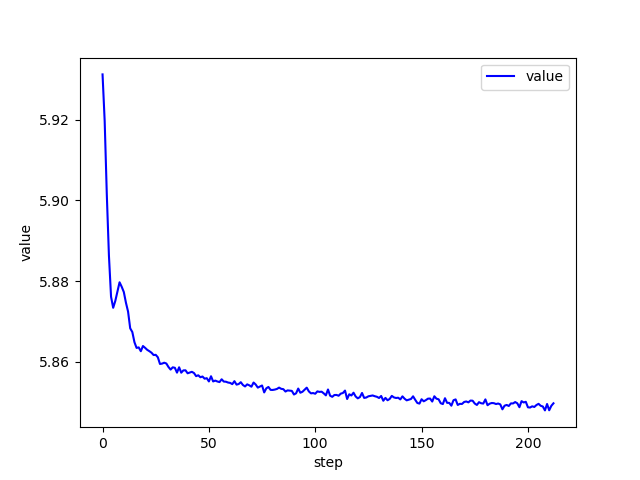
**【撰写内容】**

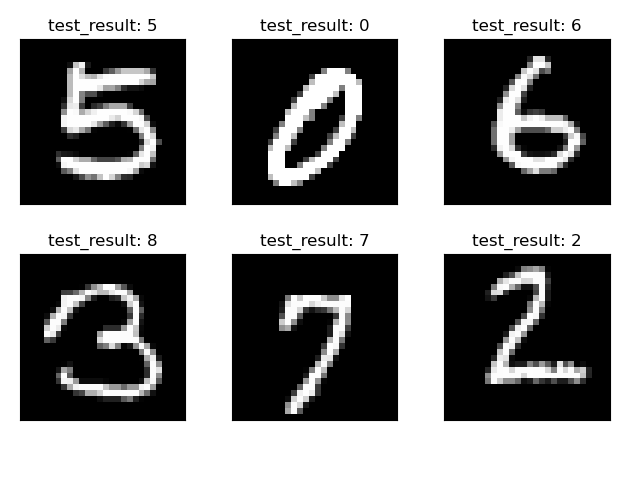
①10%样本：

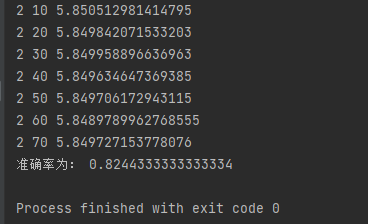


由于样本数量仅有10%，导致运行时间短且准确率较低。

②30%样本：

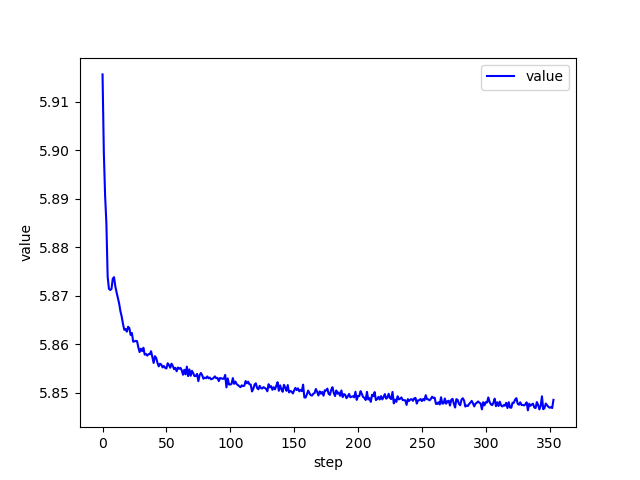
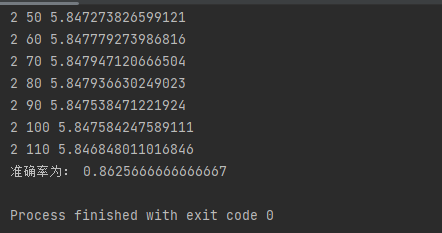
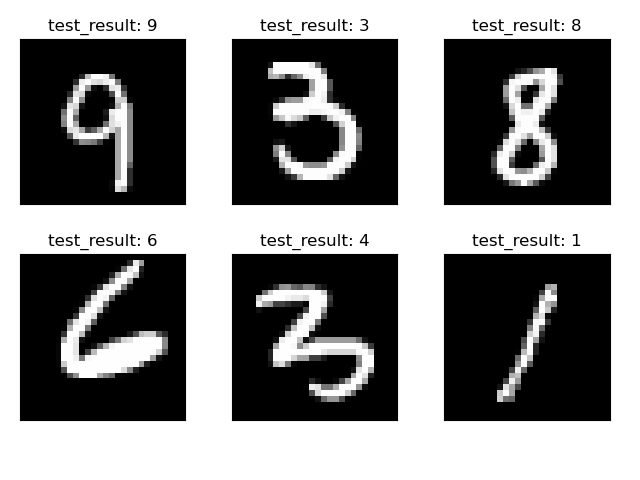






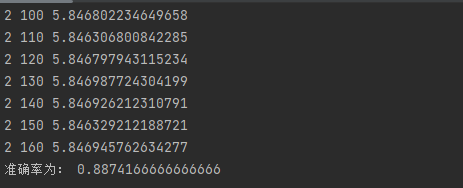
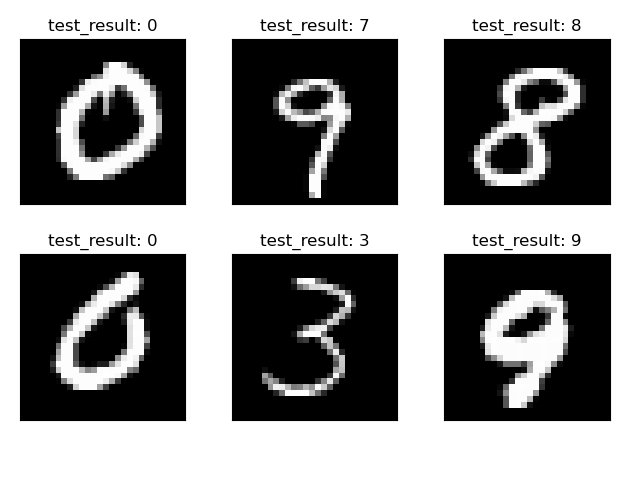
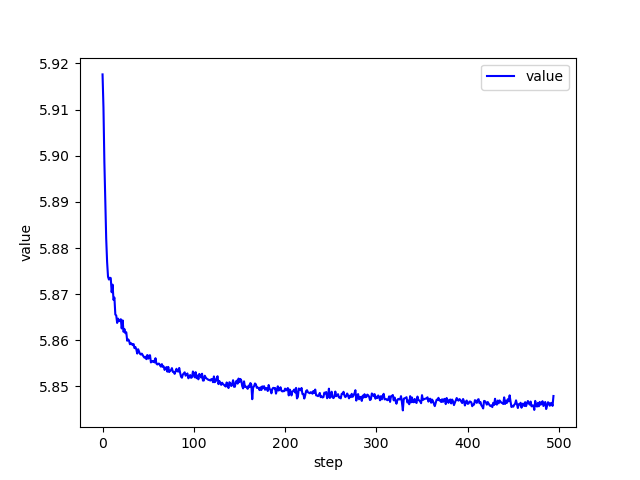
与10%相比准确率更高。

③50%样本：

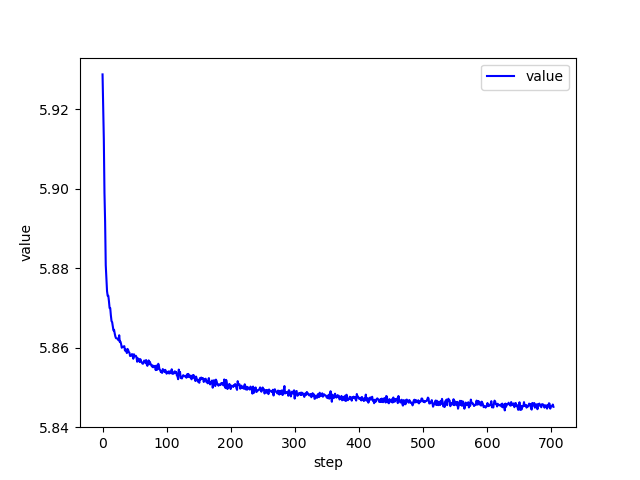
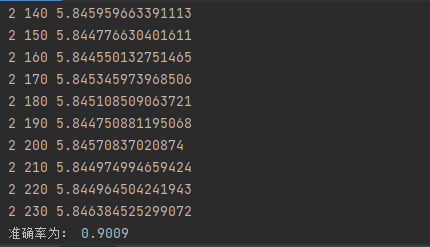
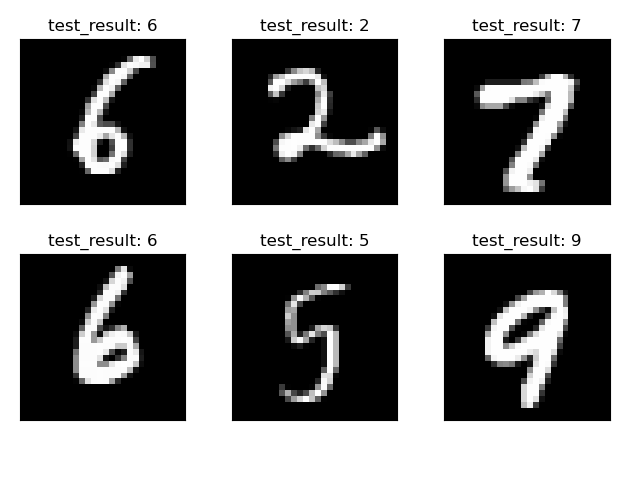
与30%相比准确率更高，但提升没有那么大，识别结果相对正确。

④70%样本：



准确率相较50%样本得到了进一步提升，但提升不明显。

⑤全样本：

准确率达到了90%以上，识别结果基本正确。

1. 实验总结。基于整个实验过程与结果，对实验进行总结。

**【撰写内容】**

本次实验目标是使用神经网络对MINST手写数字进行分类。通过三个全连接层的神经网络（卷积见后），可以达到90%以上的准确率，证明了神经网络可以有效地对手写数字进行分类。实验也表明，随着迭代次数的增加，误差下降曲线逐渐趋于平稳，说明神经网络的训练效果不断得到提高。

其中，训练集的大小一定程度上决定了模型训练的准确程度。当模型表现不佳时，可通过增加训练数据，调整模型结构等方式来进一步提高模型性能。

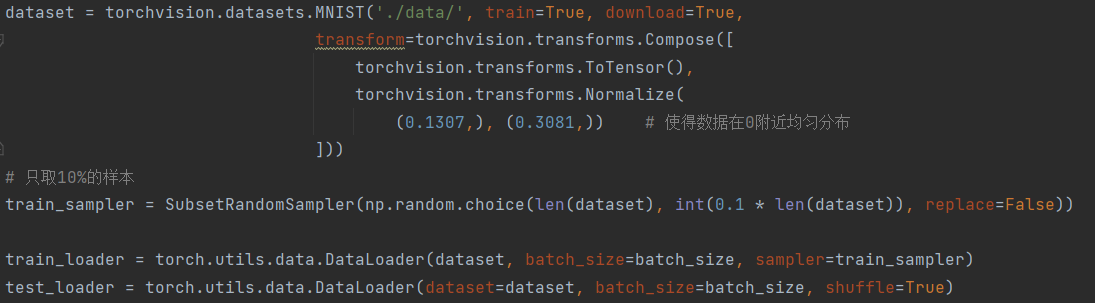
激活函数可以将线性关系转化为非线性关系，增强了神经网络的表达能力。MNIST数字识别通常采用relu函数作为激活函数。

1. 小样本分类创新实验。以10%抽样的训练集数据为训练样本，剩余的训练样本可根据需要作为无标签数据进行利用。请查阅有关课外文献，探索新的特征表示或分类模型，要求是深度神经网络类型的学习方法。包括有监督、半监督或无监督等模式均可，代表性的方法如深度卷积网络、生成式对抗网络、强化学习、Transformer、自监督学习、去噪/变分自编码器等。在完整的测试集上进行性能的评估。本部分的撰写内容自行安排，将样本的选择、模型的构建、参数的设定、训练过程及结果进行清晰的描述与展示即可。要注重实验结果的对比，结合实验结果的分析得出模型的优点与缺点，以及可能改进的方向。

**【撰写内容】**

**模型构建：**在神经网络中加入了卷积网络，构成两层卷积层和两层线性层。

训练样本(train\_loader)为10%抽样样本，测试样本(test\_loader)为全样本：



**参数设定：**

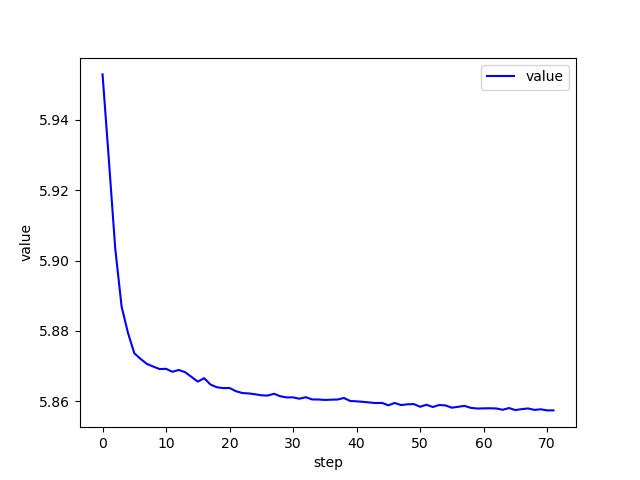
卷积核大小：5\*5，池化窗口大小：2\*2

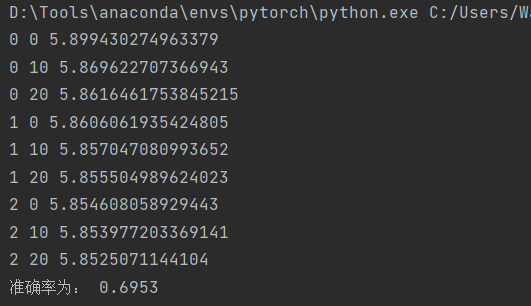
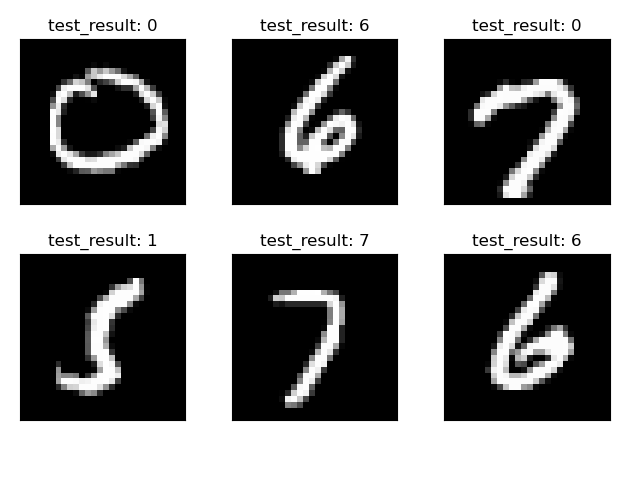
第一层卷积输入层通道为1（灰度图像以1输入），输出层为10

第二层卷积输入层通道为10，输出层为20

可得全连接层输入为1\*20\*4\*4=320，输出为10个类别

**训练三次得到的结果：**





经测试，在全样本上训练的模型，测试准确率能提高至94%以上，相较全连接模型有了进一步完善；但在10%样本的训练模型上表现不如全连接层，可能是训练样本不够限制了卷积层的作用发挥。

**优点：**准确率得到了提升，不过10%的训练样本表现不明显；同时缩小了通道尺寸，大幅减少了训练的参数数量，降低了过拟合的风险。

**缺点：**运行时间较慢。因为计算量较大，有大量的卷积操作，因此训练速度会减慢；部分特征可能被过滤掉，会存在丢失信息的问题。

**改进方向**：CNN模型结构是目前广泛使用的，可以引入更多的结构，如残差连接；特征提取主要是通过卷积、池化等操作，可以探索更多提取方式，如生成对抗网络。