智能算法与应用

实验作业

姓名：郝裕玮

班级：计科1班

学号：18329015

目录

[1 实验环境 3](#_Toc102841112)

[2 网络结构 3](#_Toc102841113)

[3 实验结果截图 5](#_Toc102841114)

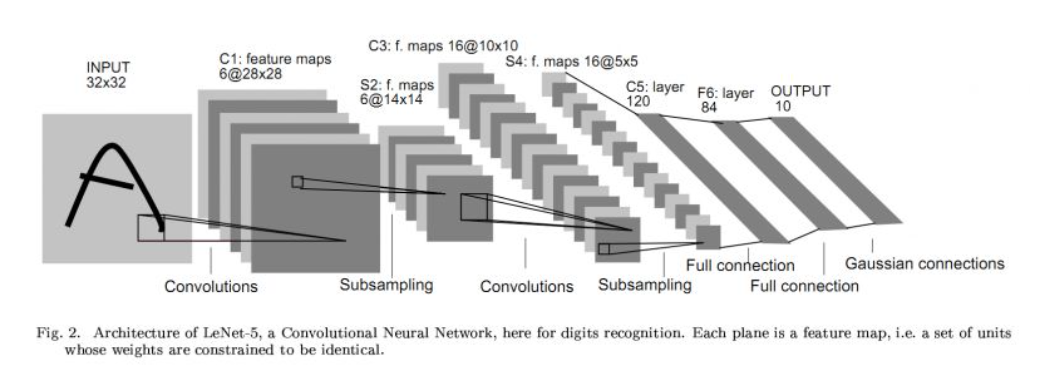
[4 实验心得 6](#_Toc102841115)

# 1 实验环境

VSCode + Python 3.8.5 + Anaconda3 + Pytorch 1.11.0

# 2 网络结构

网络结构参考了经典的LeNet-5，如下图所示：



第一层卷积层：卷积特征提取（卷积核）为5\*5，输入为（1,28,28），输出为（16,28,28）

        self.conv1 = nn.Sequential(

            # 第一个卷积con2d

            nn.Conv2d(  # 输入图像大小(1,28,28)

                in\_channels = 1,  # 输入信号的通道数：因为MINIST数据集是灰度图像，所以只有一个通道

                out\_channels = 16,  # 卷积后输出结果的通道数

                kernel\_size = 5,  # 卷积核的形状为5\*5

                stride = 1,  # 卷积每次移动的步长

                padding = 2,  # 处理边界时填充0的数量：padding = (kernel\_size-1)/2

            ),  # 输出图像大小(16,28,28)

第一层池化层：用2\*2的池化核对（16,28,28）的图片进行池化，输出为（16,14,14）的图片。

            # 激活函数

            nn.ReLU(),

            # 池化，下采样

            nn.MaxPool2d(kernel\_size=2),  # 在2x2空间下采样

            # 输出图像大小(16,14,14)

其中卷积之后使用ReLU作为激活函数，原因是增加非线性激活函数，增强拟合复杂函数的能力。

第二层卷积层：卷积特征提取（卷积核）为5\*5，输入为（16,14,14），输出为（32,14,14）

        # 建立第二个卷积(Conv2d)-> 激励函数(ReLU)->池化(MaxPooling)

        self.conv2 = nn.Sequential(

            # 输入图像大小(16,14,14)

            nn.Conv2d(  # 也可以直接简化写成nn.Conv2d(16,32,5,1,2)

                in\_channels = 16,

                out\_channels = 32,

                kernel\_size = 5,

                stride = 1,

                padding = 2

            ),

            # 输出图像大小 (32,14,14)

第二层池化层：用2\*2的池化核对（32,14,14）的图片进行池化，输出为（32,7,7）的图片。

            nn.ReLU(),

            nn.MaxPool2d(2),

            # 输出图像大小(32,7,7)

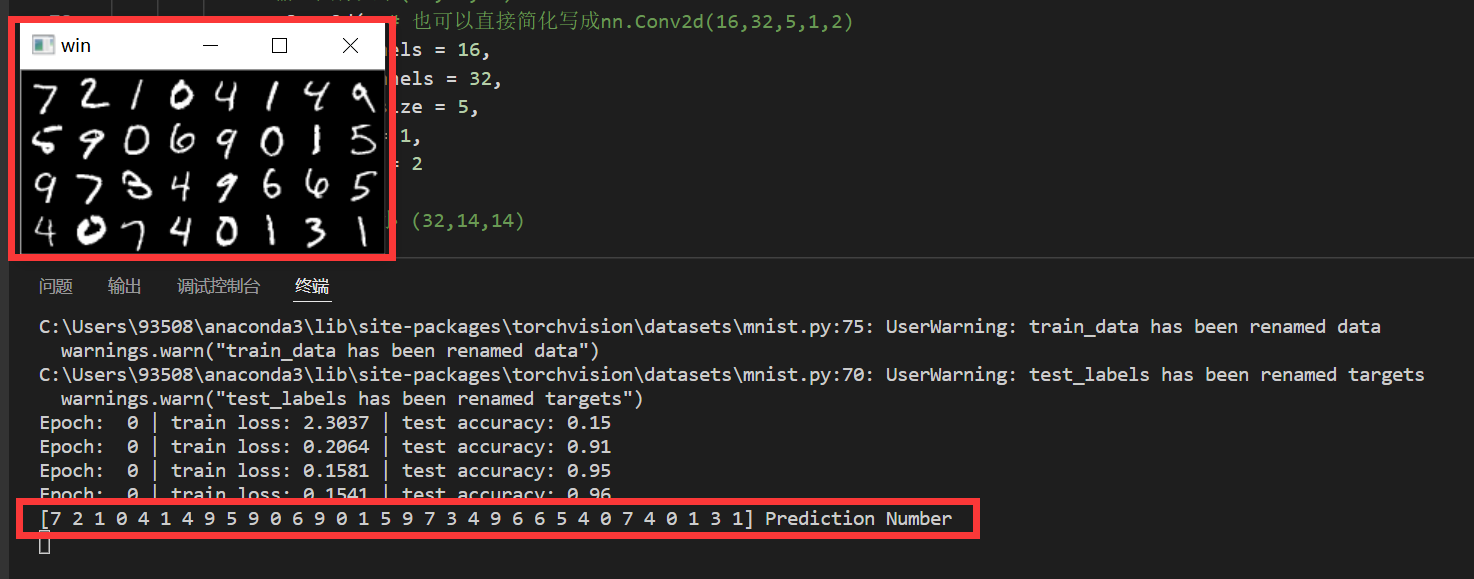
最后建立全卷积连接层，最终结果输出10个类，即预测结果。

        # 建立全卷积连接层

        self.out = nn.Linear(32 \* 7 \* 7, 10)  # 输出是10个类

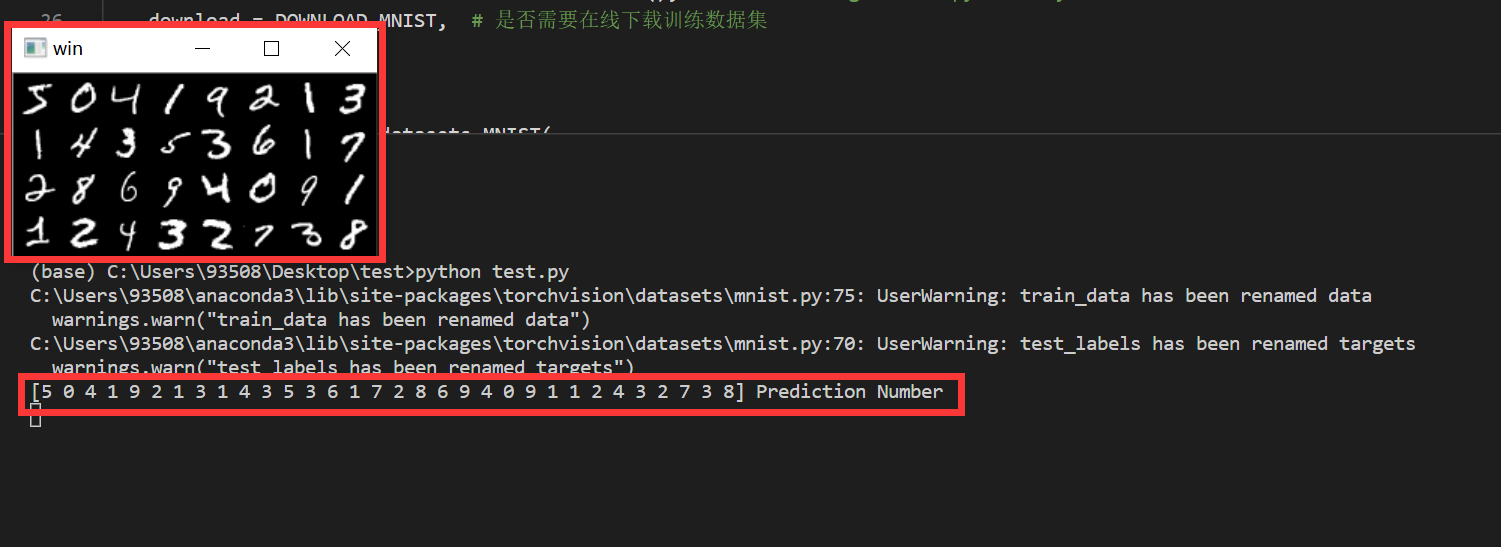
# 3 实验结果截图

训练集结果：



由上图可知测试准确率高达96%

测试集结果：



# 4 实验心得

从本次实验中我对于CNN算法有了更深刻的理解，明白了其中的原理，激发了我对机器学习的兴趣，希望在后续的学习中可以掌握更多有关神经网络的算法。