

# 中山大学计算机学院

## 人工智能本科生实验报告

课程名称: Artificial Intelligence

| 教学班级 | 网安软工合班   | 专业 (方向) | 网络空间安全 |
|------|----------|---------|--------|
| 学号   | 20337251 | 姓名      | 伍建霖    |

### 一、实验题目

#### 实验任务

- 运用pytorch框架完成中药图片分类，具体见给出的数据集和测试集。
- 需要画出loss、准确率曲线图。

#### 报告提交要求

- 提交一个压缩包。压缩包命名为：“学号\_姓名\_作业编号”，例如：20220525\_张三\_实验8。
- 压缩包包含三部分：code文件夹和实验报告pdf文件
  - Code文件夹：存放实验代码
  - Pdf文件格式参考发的模板
- 如果需要提交新版本，则在压缩包后面加\_v1等。如“学号\_姓名\_作业编号\_v1.zip”，以此类推。
- 截止日期：6月1日晚24点之前
- 提交邮箱：[zhangyc8@mail2.sysu.edu.cn](mailto:zhangyc8@mail2.sysu.edu.cn)

20

### 二、实验内容

#### 1. 算法原理

##### 1.1 主要步骤

1. 读入训练集和测试集中的数字图片信息以及对图片预处理
2. 用pytorch搭建神经网络
3. 清理上一步的梯度，然后将一个batch的训练集中的图片输入至神经网络，得到所有数字的预测分类概率
4. 根据真实标签和预测标签，利用交叉熵损失函数计算loss值，然后loss反向传播，接着更新优化器
5. 根据测试集计算准确率，如果准确率没收敛，跳转回步骤3

## 1.2 读入数据

我使用DataLoader和datasets中的ImageFolder来读入图像，resize为32乘32并标准化特征

## 1.3 CNN神经网络框架

主要有以下14层构成Conv -> BatchNorm -> ReLU -> Conv -> BatchNorm -> ReLU -> MaxPool -> Conv -> BatchNorm -> ReLU -> Conv -> BatchNorm -> ReLU -> Linear

其中Conv为卷积层，用于检测目标的特征，BatchNorm用于标准化输入，ReLU为激活函数，将所有传入的特征定义为自然数，Linear计算每个label的分数，MaxPool确保目标的位置不会影响神经网络

## 1.4 损失值和优化器

我分别使用了CrossEntropyLoss和Adam来作为损失函数和优化器

## 2. 伪代码

```
def train_test(epochs):
    device = device = torch.device("cpu")
    model.to(device)

    循环epochs次:
        从trainset中取出一个batch的图像及其对应label
        清理上一步的梯度
        输入神经网络并得到输出
        计算loss
        loss反向传播
        更新优化器
        记录loss值

    test
```

## 3.关键代码展示（带注释）

```
def train_test(num_epochs):
    # train part
    best_accuracy = 0.0
    device = torch.device("cpu")
    print("run on", device)
    model.to(device)

    for epoch in range(num_epochs):
        for i, (images, labels) in enumerate(train_loader, 0):
            images = variable(images.to(device))
            labels = variable(labels.to(device))
            optimizer.zero_grad() # 清理上一步的梯度
            outputs = model(images) # 输入神经网络并得到输出
            loss = loss_fn(outputs, labels) # 计算loss
            loss.backward() # loss反向传播
            optimizer.step() # 更新优化器
            loss_history.append(loss.item()) # 记录loss值
        accuracy = testAccuracy()
        accuracy_history.append(accuracy)
        print('epoch', epoch+1, ': accuracy is', accuracy)

    if accuracy > best_accuracy:
```

```

        best_accuracy = accuracy
    print("Best accuracy is", best_accuracy)
    print("\n---Train finished---\n")
    # test part
    images, labels = next(iter(test_loader))
    outputs = model(images)
    _, predicted = torch.max(outputs, 1)
    print('Real label: ', ' '.join('%5s' % classes[labels[j]]
                                    for j in range(10)))
    print('Predicted: ', ' '.join('%5s' % classes[predicted[j]]
                                   for j in range(10)))

```

## 4.优化

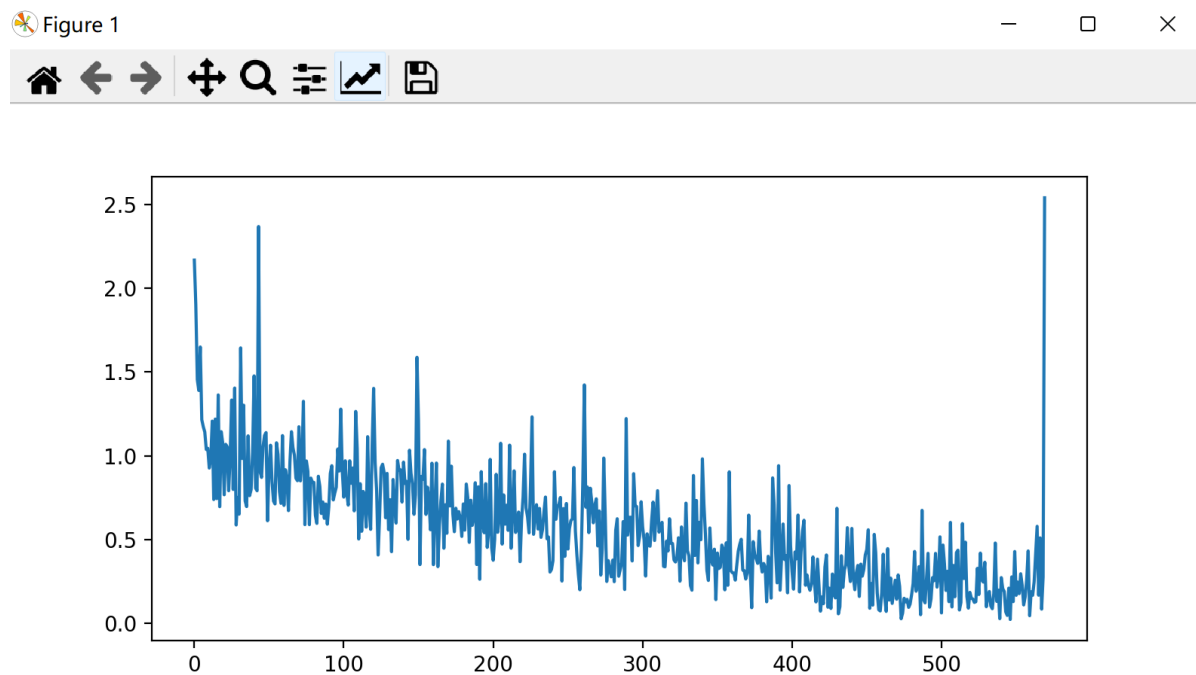
可以使用torch中的save()来保存最准确的model

## 三、实验结果及分析

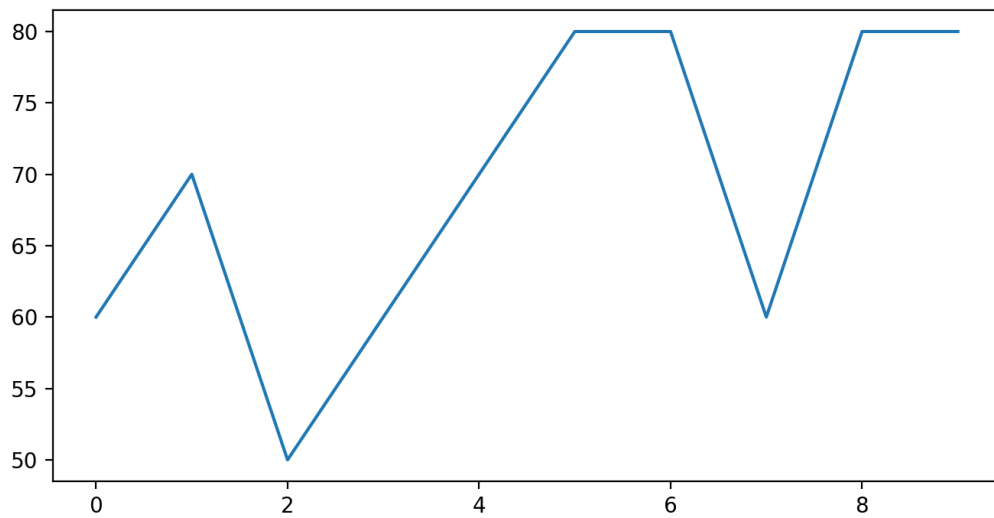
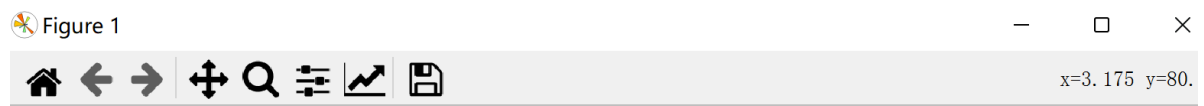
### 1. 实验结果展示示例（可图可表可文字，尽量可视化）

#### 1.1 batch=16, epoch=10

损失值

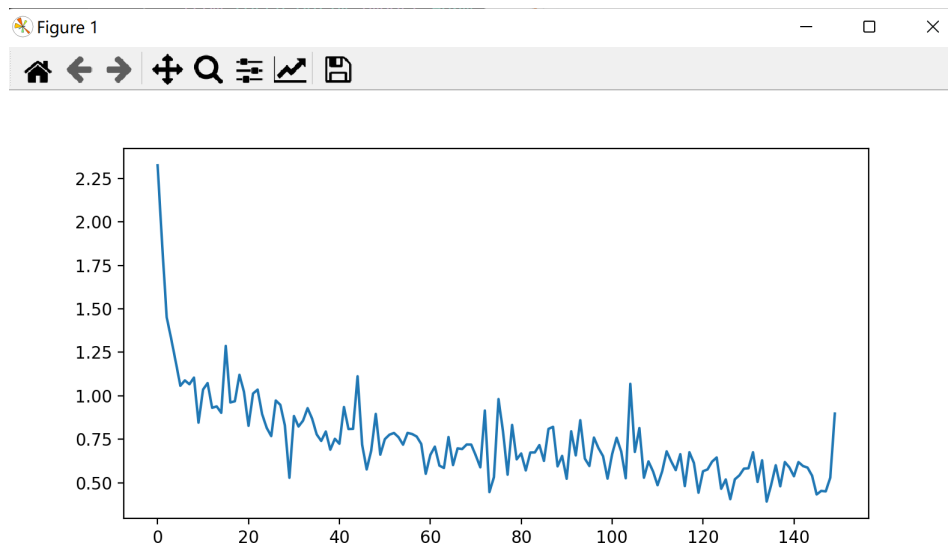


准确度

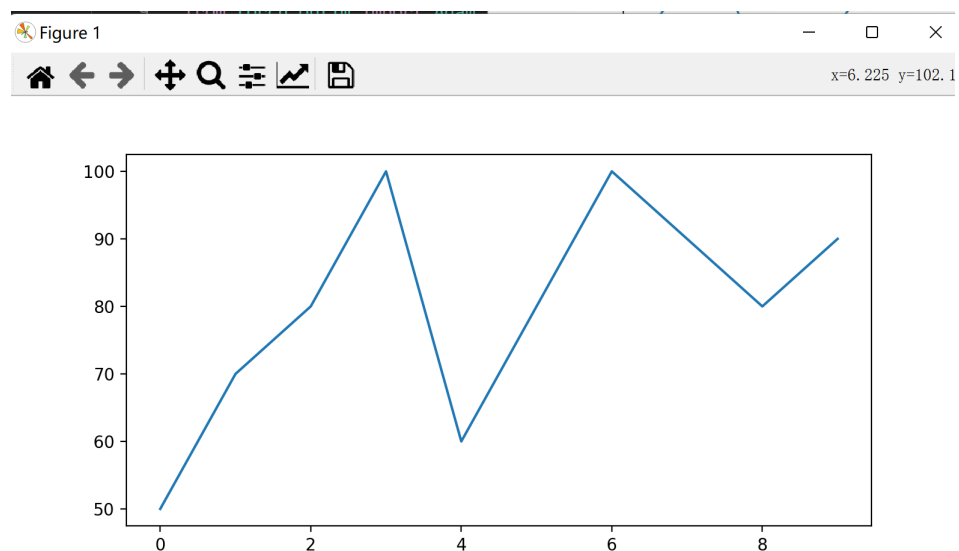


## 1.2 batch=64, epoch=10

损失值

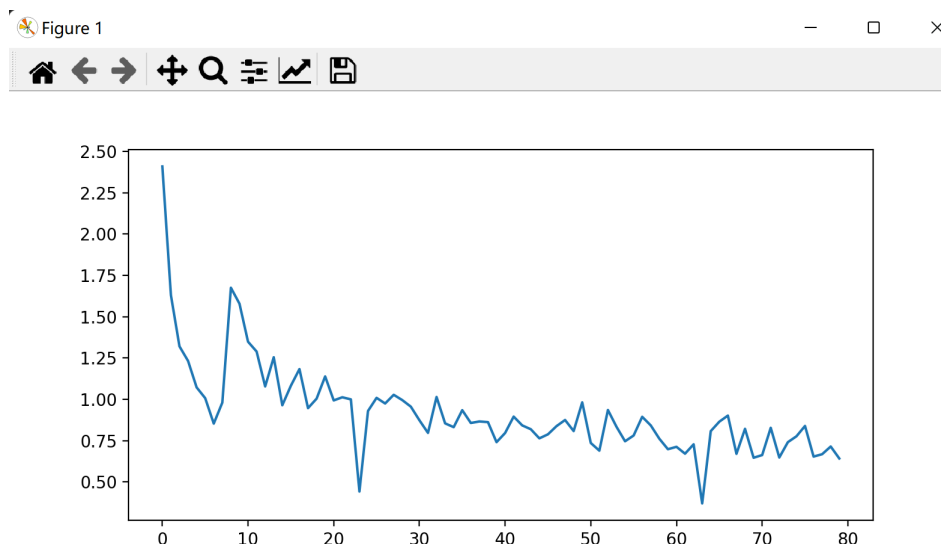


准确度

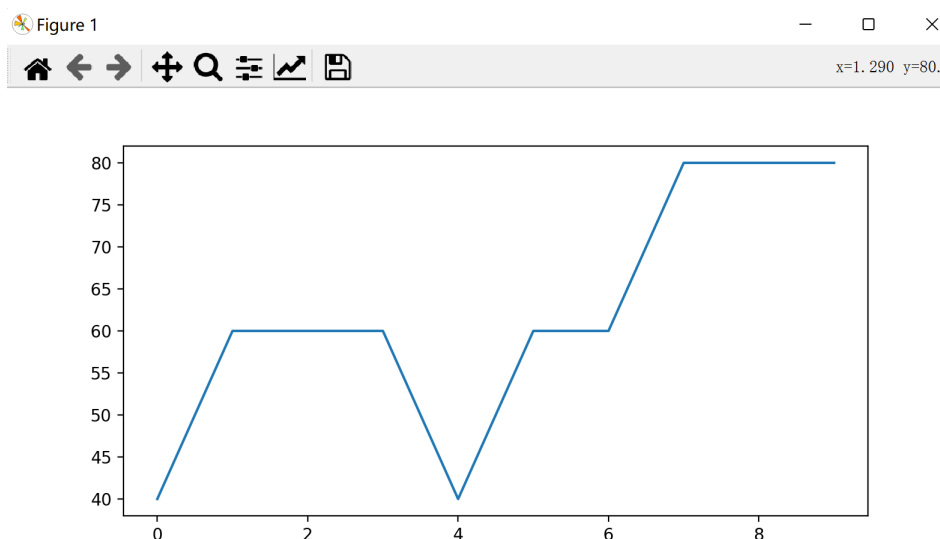


## 1.3 batch=128, epoch=10

### 损失值



### 准确度



## 2.评测指标展示及分析（其它可分析运行时间等）

当epoch都为10的情况下，batch为64时的准确度最高，4次90%以上，反而当batch为128时准确度最低，仅有三次80%（当我将其学习率调为0.0001时，准确度最高仅50%），由此可见batch越高时，学习率也因相应提高

## 四、参考资料

[Batch Norm详解之原理及为什么神经网络需要它 - 知乎\(zhihu.com\)](https://www.zhihu.com/question/26660386/answer/111111111)

[CNN中的maxpool到底是什么原理？ - 云+社区 - 腾讯云\(tencent.com\)](https://cloud.tencent.com/developer/article/1111111)

[神经网络15分钟入门！——反向传播到底是怎么传播的？ - 知乎\(zhihu.com\)](https://www.zhihu.com/question/26660386/answer/111111111)

[对Pytorch中backward \(\) 函数的理解beebabo的博客-CSDN博客.backward](https://www.csdn.net/article/1111111)

[卷积神经网络（CNN）基础及经典模型介绍 - 知乎\(zhihu.com\)](https://www.zhihu.com/question/26660386/answer/111111111)

<https://www.bilibili.com/video/BV1Vx411j7kT?p=19>

[https://github.com/MorvanZhou/PyTorch-Tutorial/blob/master/tutorial-contents/401\\_CNN.py](https://github.com/MorvanZhou/PyTorch-Tutorial/blob/master/tutorial-contents/401_CNN.py)

