



**计算机视觉**

**上机实验报告**

|  |  |
| --- | --- |
| 专 业： | 人工智能 |
| 班 级： | 192204 |
| 学 号： | 19220432 |
| 姓 名： | 陆昊宇 |

2025年03月04日

## 一、实验目的

1. 熟悉使用pytorch实现简单的分类网络。

2. 熟悉不同激活函数间的差异。

## 二、实验内容要求

#### 1. 图像分类实验

（1）构建一个LeNet-5网络，在CIFAR10数据集上完成分类任务，并给出测试结果；

（2）对比并分析ReLU、Sigmoid和tanh激活函数对收敛速度的影响，将结果可视化。

**提交入口：**

https://send2me.cn/y2QLacrr/QMiYmwT5\_S3s4g

## 三、实验报告评分标准

1. 完成数字图像处理实验（1）（50’）

2. 完成数字图像处理实验（2）（30’）

5. 实验过程记录完整，表述逻辑清晰（10’）

6. 排版工整自洽，图表说明完整（10’）

## 四、实验过程

1. 搭建网络

class LeNet5(nn.Module):

def \_\_init\_\_(self, activation):

super(LeNet5, self).\_\_init\_\_()

self.conv1 = nn.Conv2d(3, 6, 5) # CIFAR10是3通道图像

self.pool = nn.MaxPool2d(2, 2)

self.conv2 = nn.Conv2d(6, 16, 5)

self.fc1 = nn.Linear(16 \* 5 \* 5, 120)

self.fc2 = nn.Linear(120, 84)

self.fc3 = nn.Linear(84, 10)

self.activation = activation

def forward(self, x):

x = self.pool(self.activation(self.conv1(x)))

x = self.pool(self.activation(self.conv2(x)))

x = x.view(-1, 16 \* 5 \* 5)

x = self.activation(self.fc1(x))

x = self.activation(self.fc2(x))

x = self.fc3(x)

return x

1. 导入数据并预处理

# 数据预处理

transform = transforms.Compose([

transforms.ToTensor(),

transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))

])

# 加载CIFAR10数据集

trainset = torchvision.datasets.CIFAR10(

root='./data', train=True, download=False, transform=transform

)

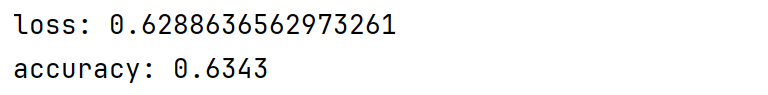
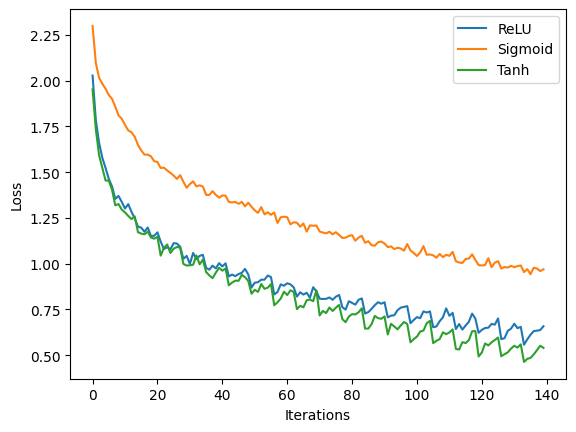
trainloader = torch.utils.data.DataLoader(trainset, batch\_size=32, shuffle=True)

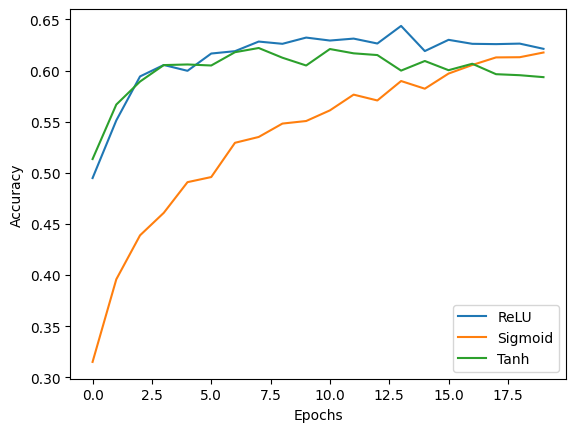
testset = torchvision.datasets.CIFAR10(

root='./data', train=False, download=False, transform=transform

)

testloader = torch.utils.data.DataLoader(testset, batch\_size=32, shuffle=False)

1. 测试模型，在CIFAR10数据集上完成分类任务  
   
2. 对比分析不同的激活函数（loss）  
   
3. 对比分析不同的激活函数（acc）



## 实验结论

从loss 图发现，Sigmoid的收敛速度不如 Relu 和 Tanh。

从 acc 图发现，Tanh相比Relu更容易过拟合，使用Relu的网络泛化性能更好。

此外，在实验中观察到Adam优化器相较于SGD优化器，对sigmoid激活函数有显著提升。而类似的情况在使用Relu和Tanh的网络中就不明显。