实验1-深度学习框架熟悉

1 实验环境

Ubuntu 20.04，Python 3.9， PyTorch1.13，Cuda11.7；

|  |
| --- |
| 1 |
| 图1 Python环境 |

2 实验过程

2.1 读取数据

|  |
| --- |
|  |
| 图2 加载数据 |

在本次实验中，所使用的是MINST数据集，其中训练集train一共包含了 60000 张图像和标签，而测试集一共包含了 10000 张图像和标签。每张图片是一个28\*28像素点的0-9的灰度手写数字图片；

在这里，使用PyTorch自带的工具包读取MINST数据集并进行装载，batch\_size设置一个批次是多少张图片，shuffle=True代表对数据集进行打乱，num\_worker代表使用多线程将数据加载进内存中；

2.2 构建模型

MLP是多层感知机结构，在PyTorch中使用nn.Linear层进行构建；在这里使用两个Linear层，在每个Linear层输出上都是用LeakReLU()激活函数。

|  |
| --- |
|  |
| 图2 网络结构 |

在本次使用中使用交叉熵作为损失函数，PyTorch中的CrossEntropyLoss()函数中已经集成了softmax，则在forward函数中不需要显式对结果做归一化处理；优化器选择Adam优化器，并且使用MultiStepLR对学习率根据epoch进行动态调整；

|  |
| --- |
|  |
| 图3 优化器和损失函数 |

2.3 训练过程

forward()函数是模型进行前向传播的过程，其中img是图片数据；但是这里使用的是Linear层进行接收，需要对img数据扁平化，即将28\*28的二维转为784的一维；同时，还需要将标签转化为对应的标签Tensor，这是一个1\*10的Tensor，其中标签序号下的值为1；然后通过模型获取图像标签数据，并使用优化器去优化网络中的参数。

|  |
| --- |
|  |
| 图3 图像数据处理 |

3 实验结果

在实验过程中，具体分析了网络层数、学习率以及学习轮次带来的具体影响。

1. 不同Epoch对模型学习带来的影响。

|  |
| --- |
| 1682912068_5_64_0.00100 |
| 1682912114_10_64_0.00100 |
| 1682912185_15_64_0.00100 |
| 1682912278_20_64_0.00100 |
| 图4 在网络层数为2时，不同Epoch的学习效果 |

从上述的图片中可以看出，随着Epoch的增加，训练集上的损失有减小的趋势，但是当到达一定轮次之后，模型的loss基本不再下降，是在一定范围内进行波动；同时，结合训练集上的正确率和测试集上的正确率曲线可以看出，虽然训练集的正确率在不断上升，甚至可以达到0.997左右，但是测试集上的正确率先升高后逐渐降低；这种情况说明，学习多个轮次，模型可以获得更多的数据，但是当学习轮次过多，模型出现过拟合的现象，就是训练集正确率在不断上升，而测试集的正确率出现了下降的趋势。

1. 不同学习率对模型学习带来的影响。

|  |
| --- |
| 1682912114_10_64_0.00100 |
| 图5 网络层数为2，学习率为0.001的效果 |
| 1682949588_10_64_0.00010 |
| 图6 网络层数为2，学习率为0.0001的效果 |

从上图的结果可以看到，当学习率过小，模型学习的速度会变慢；图5和图6同样经过了10个学习轮次，0.001学习率的训练集正确率和测试集正确率比0.0001学习率要高；除此之外，0.001学习率的loss逐渐减小，降至0.1以下的区间；而0.0001学习率的loss下降相对较慢；

1. 不同网络层数对模型学习带来的影响。

|  |
| --- |
| 1682912561_10_64_0.00051 |
| 图7 网络层数为2的效果 |
| 1682946171_10_64_0.00051 |
| 图8 网络层数为3的效果 |
| 1682945329_10_64_0.00051 |
| 图9 网络层数为4的效果 |

上图是网络层数分别为2、3、4，训练10轮之后的结果曲线；图7和图8比较可以发现，当网络层数增加的时候，模型可以学习到更多的信息，网络层数为3的模型在测试集上可以达到更高的正确率，为0.98左右；而网络层数为2的模型在测试集上只有0.966左右；除此之外，在训练集上，二者的正确率也有明显差异，明显网络更深的结构，学习效果更好。

但是比较图7和图8，后者网络层数更深，loss下降更快，loss更小，但是在量化标准上，二者没有显著的差异；这可能是由于参数量增加，导致了过拟合现象；这也意味着，网络并不是越深越好，更深的网络需要更多的信息；

4 心得与体会

PyTorch 的官方文档写得非常好，很容易理解；同时，他们提供了大量的示例代码和教程，帮助我快速上手。通过这次实验，我对使用pytorch构建一个深度学习模型的过程有了一个基本的了解，并且能够设置一些简单的对照实验，用现有的理论知识对实验现象进行说明和分析。