广东工业大学

计算机学院

《计算机视觉》

实验报告



|  |  |
| --- | --- |
| 课程名称： | 计算机视觉 |
| 实验名称： | 神经网络与深度学习 |
| 指导教师： | 赵靖亮 |
| 学生姓名： | 欧炜标 |
| 班级学号： | 3121005358 |
| 实验日期： | 2024/6/4 |
| 实验成绩： |  |

实验报告撰写要求

实验操作是教学过程中理论联系实际的重要环节，而实验报告的撰写又是知识系统化的吸收和升华过程，因此，实验报告应该体现完整性、规范性、正确性、有效性。现将实验报告撰写的有关内容说明如下：

1、 实验报告模板为电子版。

2、 下载统一的实验报告模板，学生自行完成撰写和打印。报告的首页包含本次实验的一般信息：

*  组 号：例如：2-5 表示第二班第5组。
*  实验日期：例如：05-10-06 表示本次实验日期。(年-月-日)……
*  实验编号：例如：No.1 表示第一个实验。
*  实验时间：例如：2学时 表示本次实验所用的时间。

实验报告正文部分，从五个方面（目的、内容、步骤等）反映本次实验的要点、要求以及完成过程等情况。模板已为实验报告正文设定统一格式，学生只需在相应项内填充即可。（第一部分已经填好）。

3、 实验报告正文部分具体要求如下：

一、实验目的

本次实验所涉及并要求掌握的知识点。

二、实验环境

实验所使用的设备名称及规格，网络管理工具简介、版本等。

三、实验内容与实验要求

实验内容、原理分析及具体实验要求。

四、实验过程与分析

根据具体实验，记录、整理相应命令、运行结果等，包括截图和文字说明。

详细记录在实验过程中发生的故障和问题，并进行故障分析，说明故障排除的过程及方法。

五、实验结果总结

对实验结果进行分析，完成思考题目，总结实验的心得体会，并提出实验的改进意见。

注：本部分仅为展示报告内容，具体报告内容见下一页。

一、实验目的

*本次实验所涉及并要求掌握的知识点。（可多页）*

 了解、掌握神经网络、反向传播、深度卷积网络的基本原理。掌握MindSpore的安装过程，掌握使用MindSpore进行简单卷积神经网络的搭建。

二、实验环境

**硬件环境**

设备名称 LAPTOP-BHF8HOH1

处理器 11th Gen Intel(R) Core(TM) i7-11800H @ 2.30GHz 2.30 GHz

机带 RAM 16.0 GB (15.8 GB 可用)

设备 ID 8A9CD625-D5C1-40B0-AFDA-7C0219502457

产品 ID 00342-36199-79021-AAOEM

系统类型 64 位操作系统, 基于 x64 的处理器

笔和触控 没有可用于此显示器的笔或触控输入

**软件环境**

python3.12

opencv

numpy

mindspore

**IDE**

pycharm

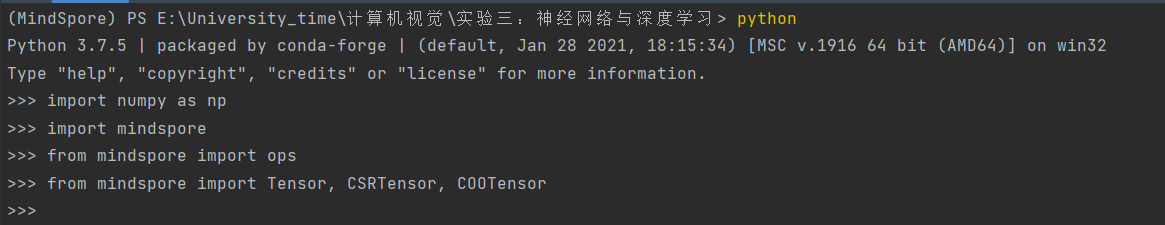
三、实验内容与实验要求

实验前理论课应该完成high-level视觉部分的讲解，学生已了解、掌握神经网络、反向传播、深度卷积网络的基本原理。本实验介绍了Window系统下的深度学习环境—华为MindSpore—的搭建和使用。

四、实验过程与分析

由于人工智能班级已经在大二的时候完成mindspore先修课，mindspore环境的配置在此不再重复。

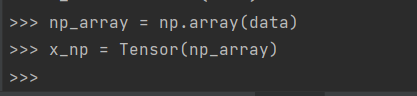
1. **mindspore中张量的定义与使用**
2. 导入对应的库



1. 由数据直接生成张量



1. 有numpy直接生成



1. 使用init函数直接生成

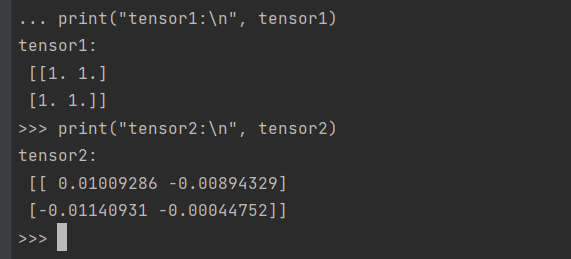
from mindspore.common.initializer import One, Normal

# 初始化一个全1的tensor

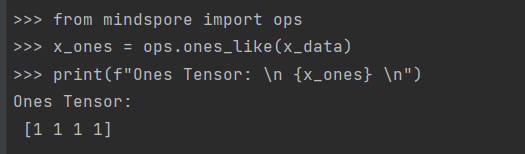
tensor1 = mindspore.Tensor(shape=(2, 2), dtype=mindspore.float32, init=One())

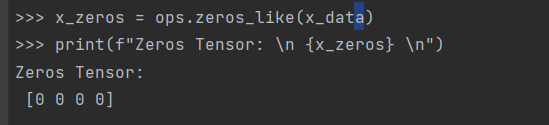
# 初始化一个符合标准正态分布的Tensor

tensor2 = mindspore.Tensor(shape=(2, 2), dtype=mindspore.float32, init=Normal())

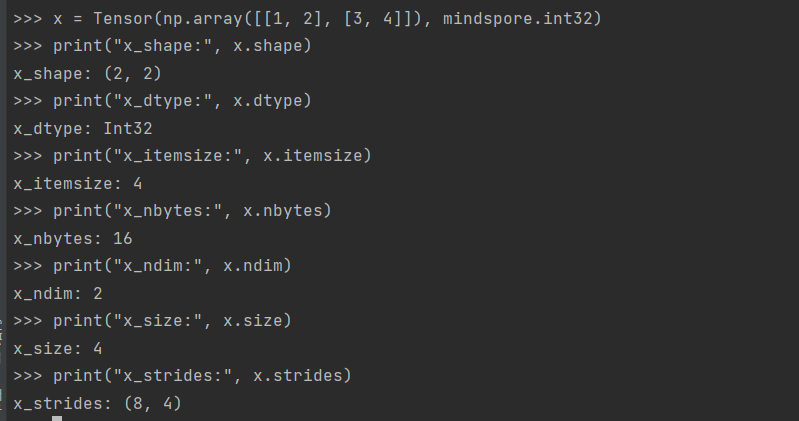


1. 继承另外一个Tensor的大小、类型，例如申请全零和全1张量时

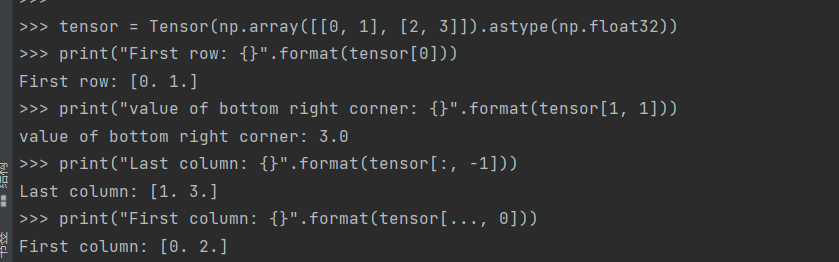




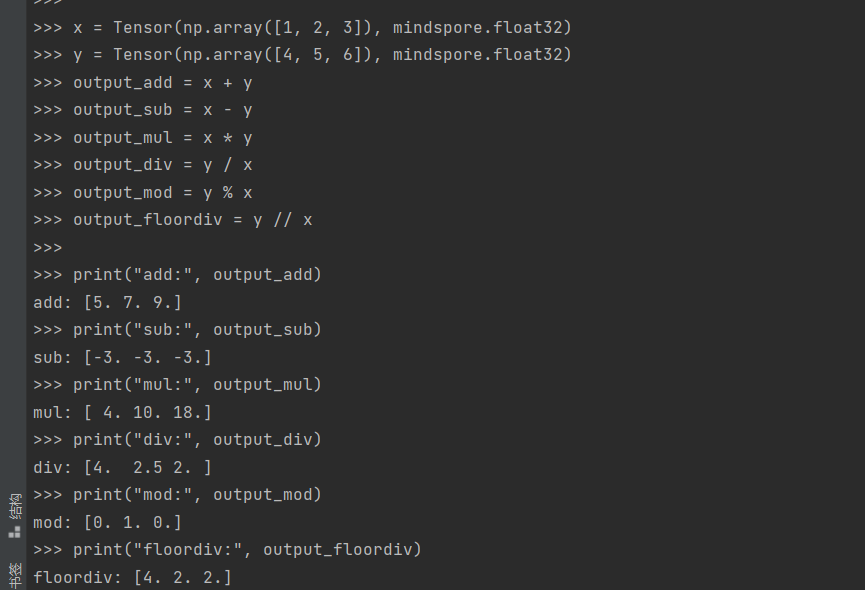
1. 查看张量属性



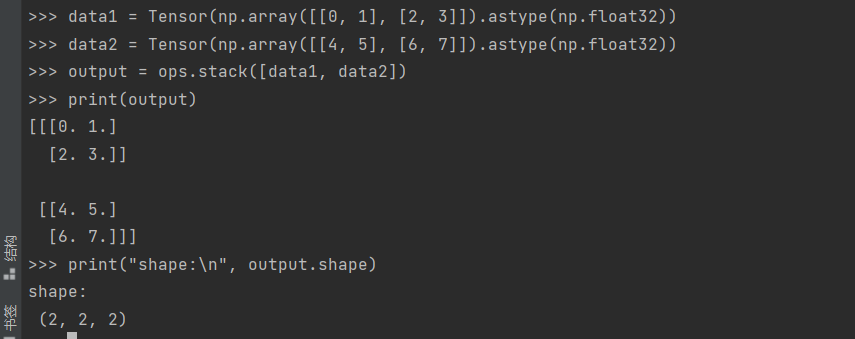
1. 张量如何索引



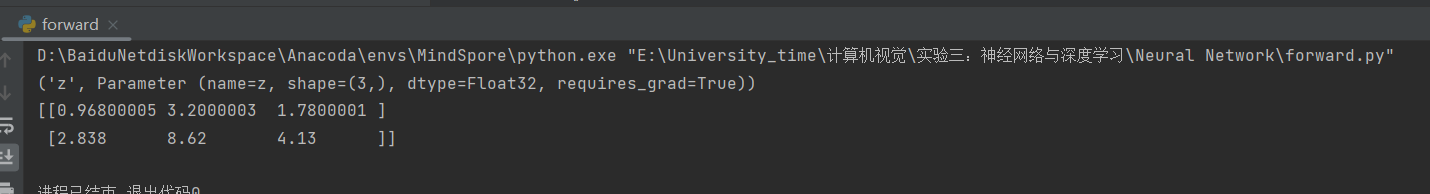
1. 张量运算



1. 其他张量运算：Tensor级联concat：通过指定具体的维度，进行级联；Tensor的堆叠



1. **前向传播**
2. 运行结果



1. class Net(nn.Cell) 功能分析

Net 类定义了一个使用 MindSpore 框架的神经网络组件，该类继承自 nn.Cell，这是 MindSpore 中定义神经网络组件的基础类。

首先进行初始化：

* super(Net, self).\_\_init\_\_() 调用父类 nn.Cell 的初始化函数。
* self.matmul = ops.MatMul() 初始化一个矩阵乘法操作。
* self.z = Parameter(Tensor(np.array([1.0, 1.0, 1.0], np.float32)), name='z') 初始化一个参数 z，它包含三个元素且值都为 1.0 的一维张量。这个参数包裹在一个 Parameter 对象中，是可训练的。

然后使用construct函数 定义了网络的前向传播

def construct(self, x, y):

x = x \* self.z

out = self.matmul(x, y)

return out

前向传播的具体过程

x = x \* self.z

out = self.matmul(x, y)

return out

1. 代码注释

import numpy as np

import mindspore.nn as nn

import mindspore.ops as ops

from mindspore import Tensor

from mindspore import ParameterTuple, Parameter

from mindspore import dtype as mstype

class Net(nn.Cell):

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

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

# 初始化矩阵乘法操作

self.matmul = ops.MatMul()

# 初始化一个参数 z，值为 [1.0, 1.0, 1.0]

self.z = Parameter(Tensor(np.array([1.0, 1.0, 1.0], np.float32)), name='z')

def construct(self, x, y):

x = x \* self.z

out = self.matmul(x, y)

return out

# 实例化 Net 类，创建模型对象

model = Net()

for m in model.parameters\_and\_names():

print(m)

# 创建 Tensor 对象 x 和 y，用于输入模型

x = Tensor([[0.8, 0.6, 0.2], [1.8, 1.3, 1.1]], dtype=mstype.float32)

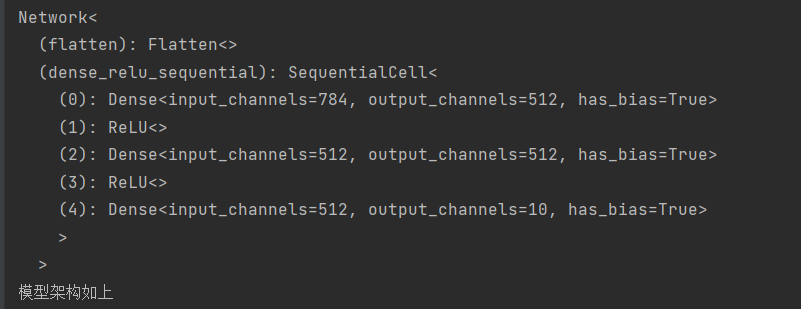
y = Tensor([[0.11, 3.3, 1.1], [1.1, 0.2, 1.4], [1.1, 2.2, 0.3]], dtype=mstype.float32)

# 使用模型进行前向传播，并输出结果

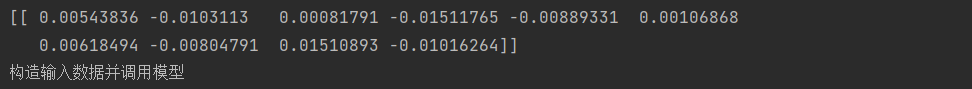
result = model(x, y)

print(result)

1. **网络构建**
2. 步骤运行结果
3. 首先定义模型的类，实例化模型并查看模型的架构



1. 构造输入数据并调用模型



1. 使用Softmax层获得预测概率



1. 分解神经网络模型的每一层并观察效果



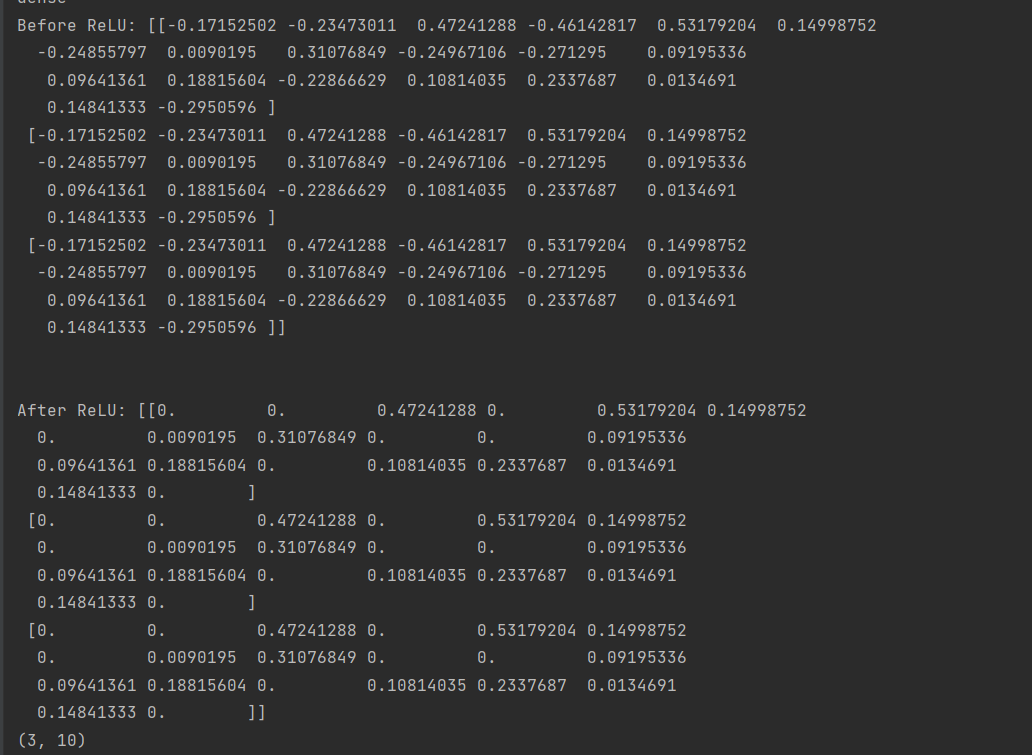
1. Flatten层



1. Dense层



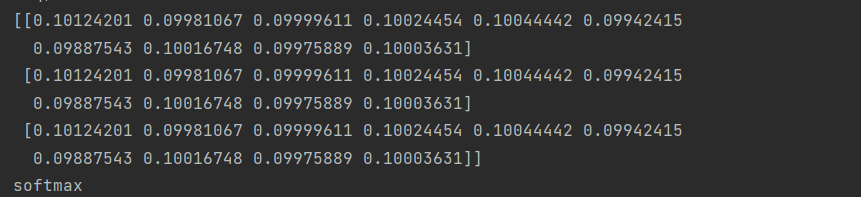
1. ReLU层



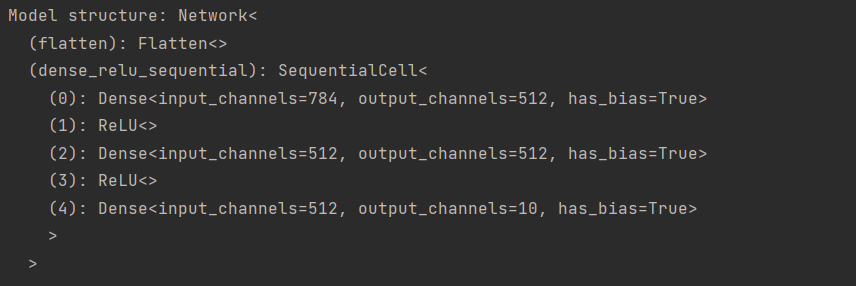
1. SequentialCell

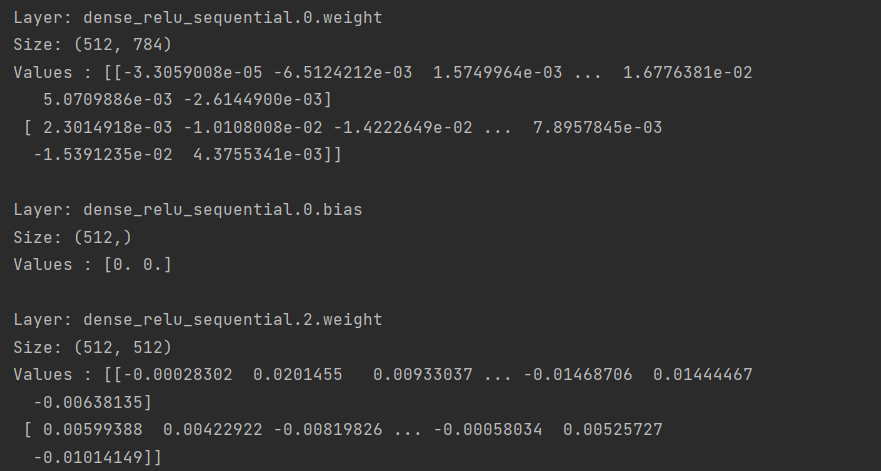


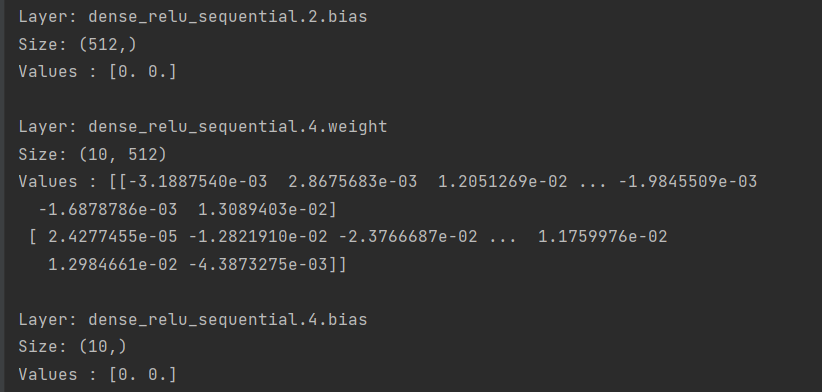
1. Softmax层



1. 模型结构以及相关参数







1. 问题解答
2. 代码第35行，nn.Dense(2828, 512) 中的 2828是什么含义？

28\*28 表示输入层的大小，因为每个 MNIST 图像是 28x28 像素，在输入到全连接层之前需要将其展平成一维张量，这个张量的大小是 784（即 28\*28）。

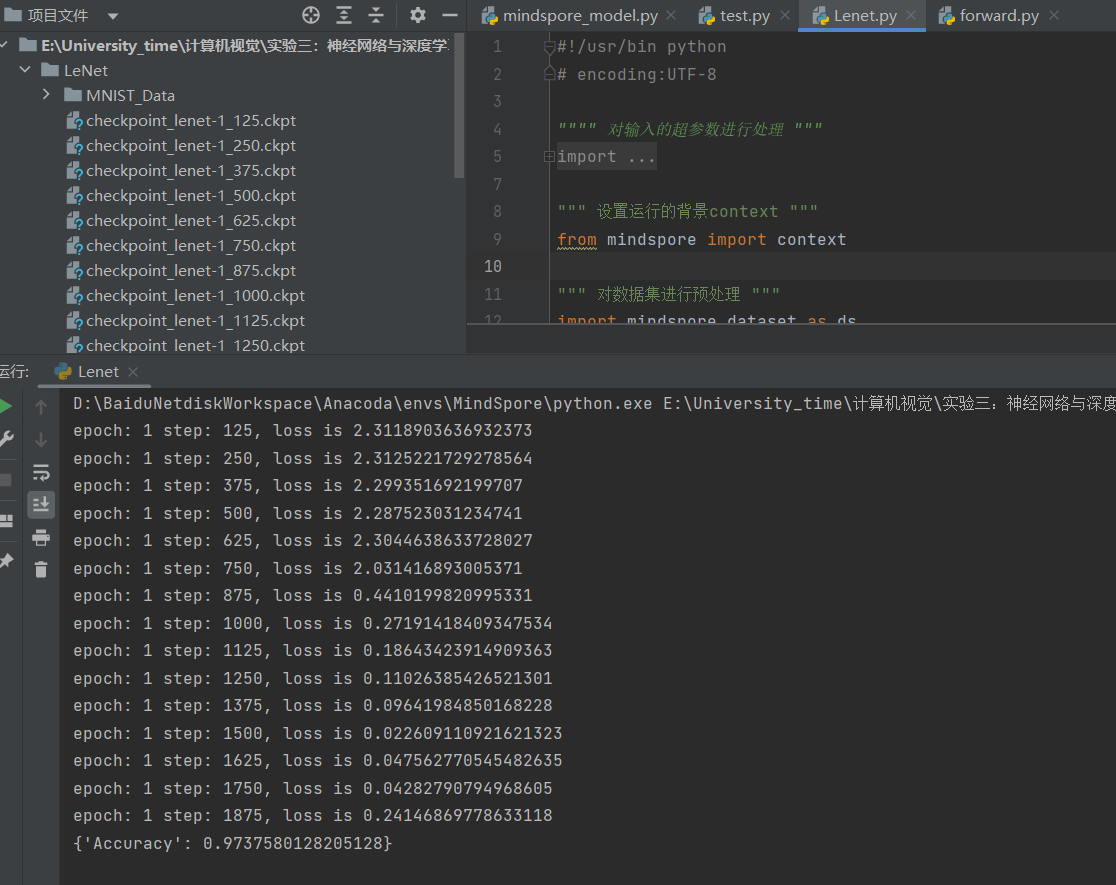
1. 代码第39行，(512, 10) 的含义是什么？

(512, 10) 表示全连接层的输入大小为 512，输出大小为 10。这个层将 512 维的输入向量转换为 10 维的输出向量，输出的 10 维向量对应于 10 个类别的预测值。

1. 第60行，(1, 28, 28) 的含义是什么？

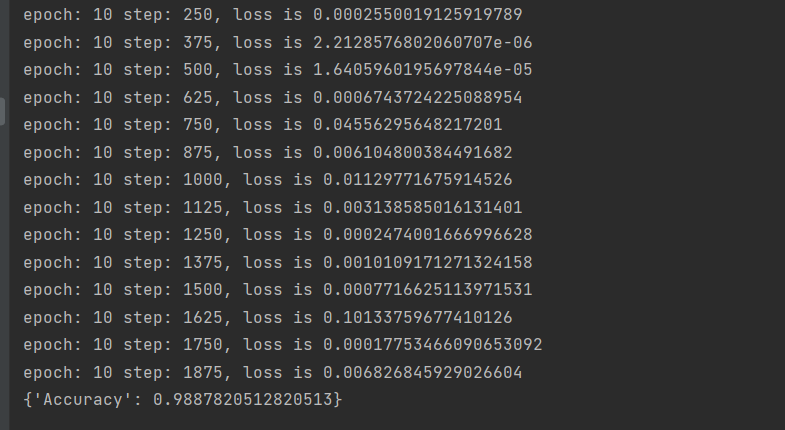
(1, 28, 28) 表示输入张量的形状，其中 1 是批量大小（即一个样本），28, 28 是每个样本的高度和宽度。这个形状表示单个 28x28 的图像。

1. **LeNet手写字体识别**
2. 运行结果



1. 修改 Lenet.py，增加训练过程的 epoch，从而使测试精度达到0.96以上。





训练10代后，正确率达到了98%以上。

1. 问题解答
2. 解释 Lenet.py 代码第110行中 sparse 参数 和 reduction 参数的含义。

**sparse** 参数含义: 当 sparse=True 时，标签应该是稀疏表示，即类别索引（整数形式）。例如，如果有3个类别，标签可以是 0、1 或 2。当 sparse=False 时，标签应为密集表示（独热编码），即每个类别都有一个独热向量。这里使用稀疏表示，能更节省内存并且计算更高效，尤其是对于多类别分类任务。

**reduction** 参数含义: 指定如何计算批量数据的损失值。包括以下三类值：

mean：对所有样本的损失取平均值。

sum：对所有样本的损失取总和。

none：不进行任何归约，返回每个样本的损失。

1. 解释第113行 momentum 参数的含义。

**momentum** 是动量参数，用于加速梯度下降优化并抑制震荡。动量的取值通常在 0 到 1 之间。一个常见的值是 0.9，表示在计算当前梯度时，会考虑上一时间步的梯度，从而在参数更新时引入动量。

1. 解释第132行 dataset\_sink\_mode 参数的含义。

**dataset\_sink\_mode** 参数含义: 控制数据是否使用数据下沉模式。dataset\_sink\_mode=True 时，数据将直接从数据集加载到设备内存中，从而提高数据处理和训练速度。dataset\_sink\_mode=False 时，数据将逐步加载到设备中，适合小数据集或调试模式。

五、实验结果总结

本次实验的主要目的是通过实践操作加深对神经网络、反向传播及深度卷积网络基本原理的理解，并掌握MindSpore框架的使用。实验中，我们成功在Windows系统上搭建了深度学习环境，并利用MindSpore框架完成了简单卷积神经网络的构建。

在实验的初始阶段，首先熟悉了MindSpore中张量的定义与使用，包括从数据直接生成张量、使用NumPy生成张量以及利用init函数直接生成张量。这些操作使我们对张量的属性、索引和运算有了更深入的认识，为后续的模型构建和训练打下了坚实的基础。

在网络构建部分，我们定义了一个简单的神经网络模型Net，并进行了前向传播的测试。通过实例化模型对象并输入Tensor对象x和y，我们观察了模型的前向传播过程，并成功输出了预期的结果，验证了模型的正确性。

我们对LeNet手写字体识别模型进行了实验。通过增加训练过程中的epoch数量，使测试精度达到了98%以上。

总的来说，本次实验不仅加深了我们对深度学习理论的理解，而且提升了我们使用MindSpore框架进行模型构建和训练的实践能力。