mindspore可视化实验报告

1. 实验目标

1.使用用一种神经网络可视化工具,并写一篇试用报告。(如Tensorflow Playground、CNN Explainer等)

2.实现LeNet-5在MNIST数据集上的训练和测试,并进行分析。

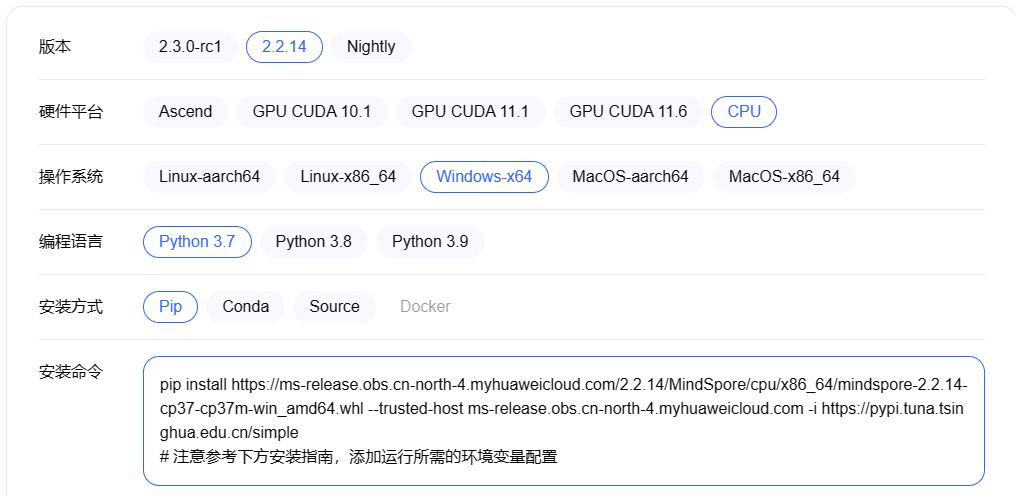
2.使用工具

使用mindspore中的可视化工具mindinsight进行试验，实现LeNet-5在MNIST数据集上的训练和测试。

1. 实验内容

### 3.1环境的安装

### 首先创建一个虚拟环境。安装Python3.7或者3.9!!其他版本的python使用mindinsight会出现报错。按照官网的方法，在window上使用pip 安装，配置命令如下：



之后还需要安装mindvision 和mindinsight，使用pip进行安装，安装完成后，便可以进行代码的编写！

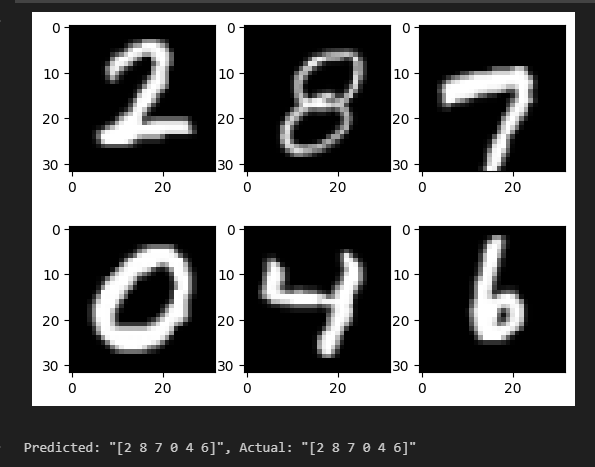
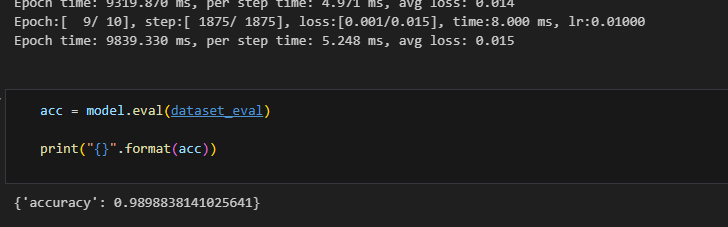
最终在window上报错太多，在ubuntu 上可以正常运行。

### 3.2网络的训练和测试

为了实现在在MNIST数据集训练或者测试LeNet-5，需要分步骤进行。

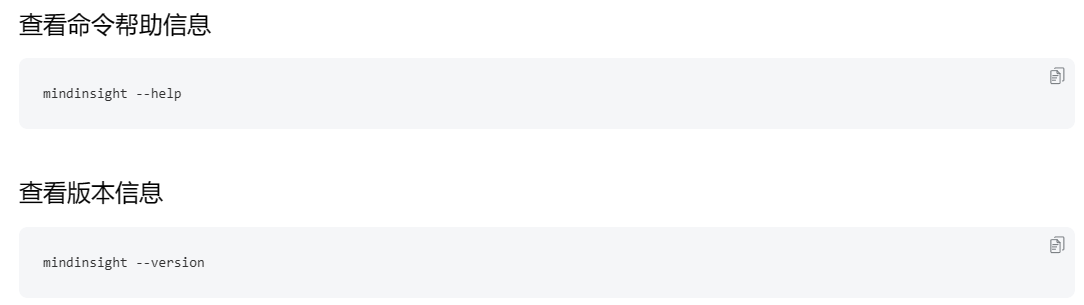
1. 下载mnist数据
2. 定义lenet网络模型
3. 定义损失函数和优化器
4. 设置模型保存参数
5. 初始化模型，并且进行训练
6. 加载模型权重，进行评估。

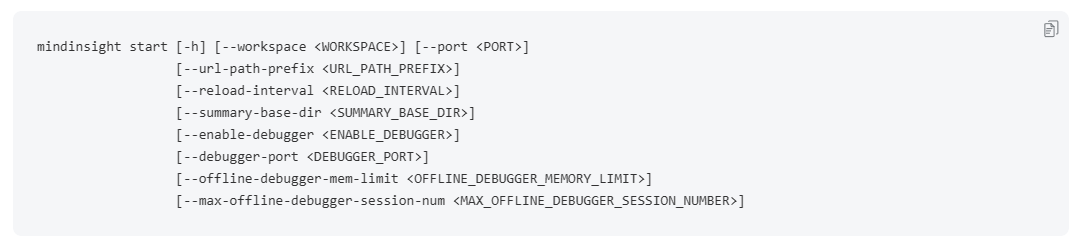
上面步骤的代码都已经写到文件中，这里就不列出来了。最终得出模型的准确率为98.98，之后进行案例的测试：



### 3.3mindinsight的可视化

Mindinsight的相关命令







### Mindinsight通过SummaryCollector自动收集

### 在MindSpore中通过 Callback 机制提供支持快速简易地收集一些常见的信息，包括计算图，损失值，学习率，参数权重等信息的 Callback，叫做 SummaryCollector。在编写训练脚本时，仅需要实例化 SummaryCollector，并将其应用到 model.train或者 model.eval中， 即可自动收集一些常见信息。

所以只需要在之前的代码中添加如下代码，可以查看训练过程。

##Collector landscape information

    summary\_collector = ms.SummaryCollector(

        summary\_dir,

        keep\_default\_action=True,

        collect\_specified\_data={

            "collect\_landscape": {

                "landscape\_size": 10,

                "unit": "epoch",

                "create\_landscape": {"train": True, "result": True},

                "num\_samples": 512,

                "intervals": [interval\_1, interval\_2],

            }

        },

        collect\_freq=1,

    )

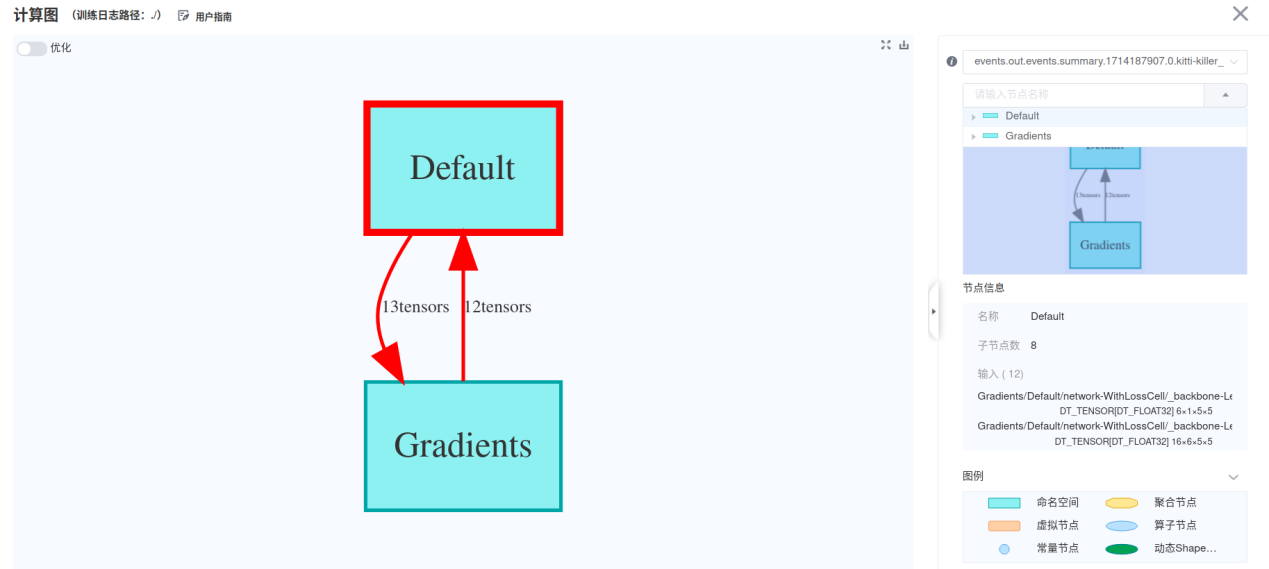
如需要查看可视化的日志，通常使用命令



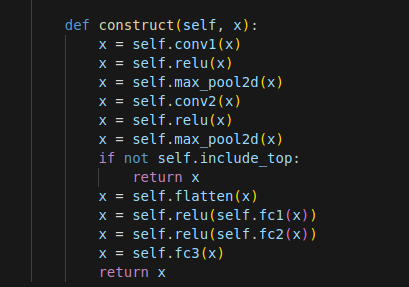
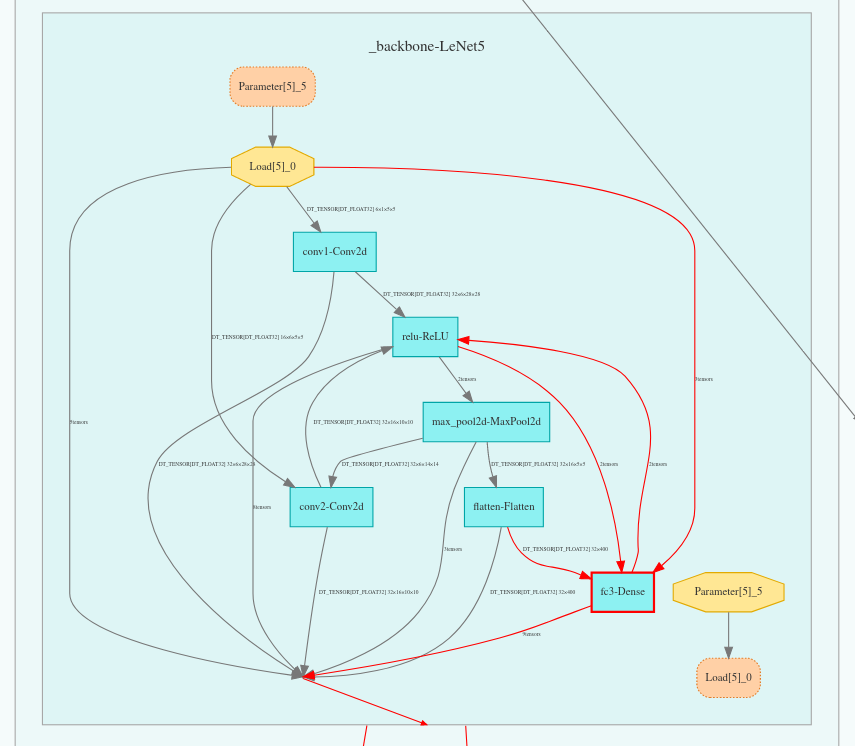
最终可视化的界面如下：



其中，计算图中显示了多种信息，包括整个网络的结构。



打开network节点，就会显示出网络的整个框架结构，可以与代码中的网络互相对应。



同时，还能展示数据图，参数的分布以及其他损失函数的训练变化。

