# 人工智能大作业报告

# 1. 介绍

## a. 手写数字识别

LeNet5 + MNIST被誉为深度学习领域的“Hello world”。本实验主要介绍使用MindSpore在MNIST手写数字数据集上开发和训练一个LeNet5模型，并验证模型精度。

## b. 垃圾分类识别

随着人们对于自然的保护，垃圾处理成为人们日常生活中必不可少的一步。深度学习计算中，从头开始训练一个实用的模型通常非常耗时，需要大量计算能力。常用的数据如OpenImage、ImageNet、VOC、COCO等公开大型数据集，规模达到几十万甚至超过上百万张。网络和开源社区上通常会提供这些数据集上预训练好的模型。大部分细分领域任务在训练网络模型时，如果不使用预训练模型而从头开始训练网络，不仅耗时，且模型容易陷入局部极小值和过拟合。因此大部分任务都会选择预训练模型，在其上做微调（也称为Fine-Tune）。本实验以MobileNetV2+垃圾分类数据集为例，主要介绍如在使用MindSpore在CPU/GPU平台上进行Fine-Tune。

# 2. 原理介绍

## a. 残差神经网络

## b. 学习率衰减策略

### ⅰ. 指数衰减

### ⅱ. 余弦衰减

### ⅲ. 多项式衰减

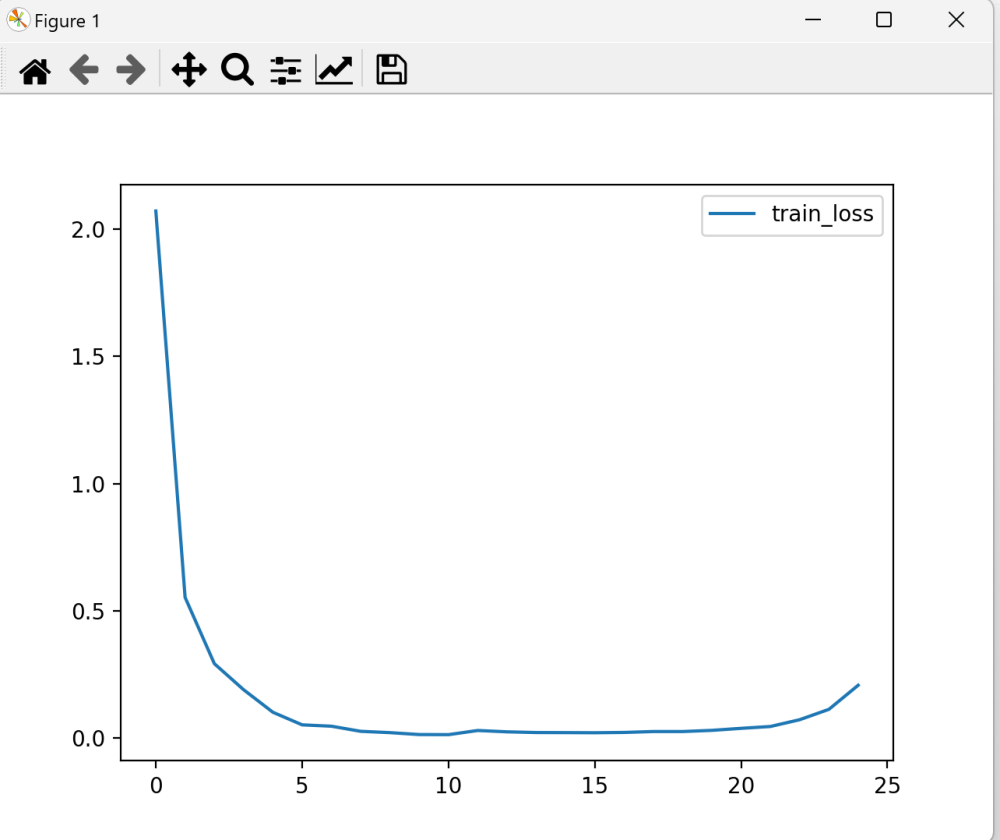
# 3. 具体实现

## a. 垃圾分类的具体调参过程

### ⅰ. epochs

迭代步数，生成模型的数量，关系到模型的拟合问题。该值过高会导致过拟合，过低会导致欠拟合

可以根据loss曲线进行分析。

如下图

可以很明显的发现，超过20之后，loss曲线开始上升，说明过拟合，需要降低。25发现开始过拟合，调节参数，但后面发现该参数对于准确度没有实质性的提高

### ⅱ. decay\_type

该参数选择学习率衰减方式，有三种衰减方式，分别是cosine、suqare和exponential。本小组的创新点在于使用学习率指数衰减的方式，指数衰减的公式为





具体实现代码如下

elif decay\_type == "exponential":  
 exponential\_decay = (0.4)\*\*(total\_steps/decay\_steps)  
 lr = (lr\_max - lr\_end) \* exponential\_decay + lr\_end

衰减率是一个可以调节的参数，本小组经过调节发现，参数在0.3-0.5比较合适，最终确定参数为0.4

### ⅲ. momentum

该参量直接使用实验报告中的推荐参数0.9即可，效果较为良好，将本小组的准确度在初始阶段有了较大的提高

### ⅳ. weight\_decay

对于准确度的提升有较大的帮助，使准确度迅速提高。weight\_decay尽可能小会比较合适。

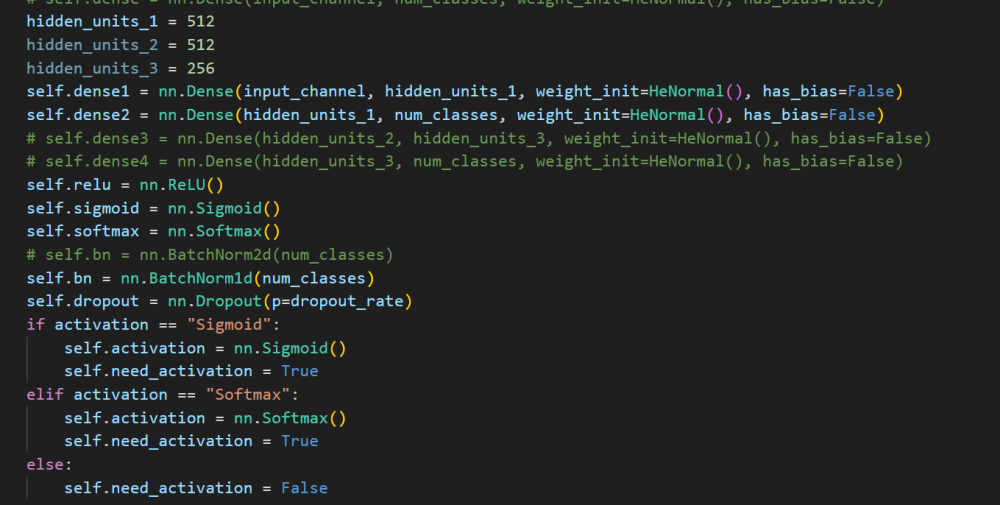
### ⅴ. reduction

reduction 池化方式有mean和max两种方式，

### ⅵ. 失败的尝试

#### 1. 添加全连接层

加入全连接层

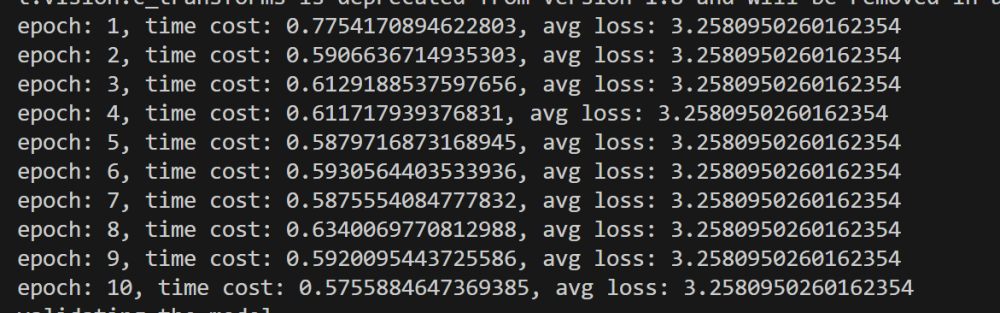


发现loss也保持不变不下降，未实现收敛

#### 2. 手动加入激活函数

将def \_\_init\_\_(self, input\_channel=1280, hw=7, num\_classes=1000, reduction='mean', activation="none"):

中的activation 改为 相应的激活函数，发现loss保持不变

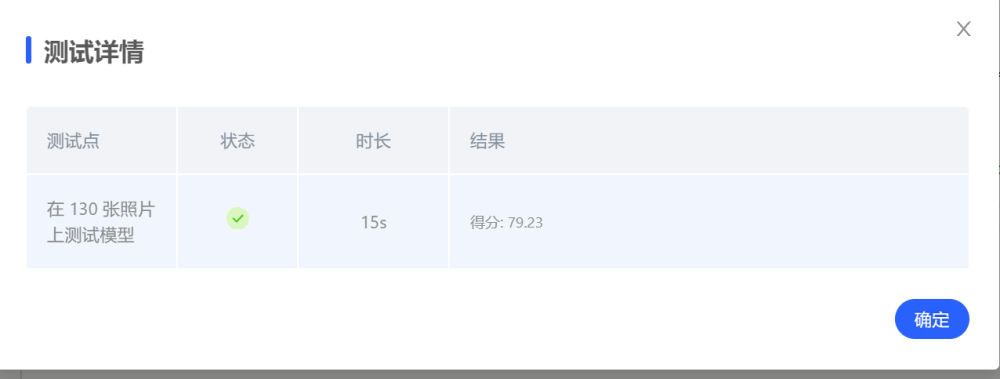


尝试失败

### ⅶ. 更改模型评估的标准

在多次尝试后发现，虽然acc可以达到很高的值，但是在130中的提交的分数在76到79徘徊，说明模型的评价的方式出现偏差，需要重新定义模型的评估标准。

# 4. 结果



# 5. 误差分析