**华中科技大学**

**计算机科学与技术学院**

**《机器学习》结课报告**



专 业： 计算机

班 级： 本硕博

学 号： U202115674

姓 名： 姚晨炫

成 绩：

指导教师： 何琨

**完成日期：2023 年 5 月 15 日**

目录

[1. 实验要求 2](#_Toc135038855)

[2. 算法设计与实现 2](#_Toc135038856)

[1.华为mindspore手写数字识别 2](#_Toc135038857)

[2.华为mindspore鸢尾花分类问题 3](#_Toc135038858)

[3. kaggle手写数字分类比赛 4](#_Toc135038859)

[4. 实验环境与平台 6](#_Toc135038860)

[5. 结果与分析 6](#_Toc135038861)

[5. 个人体会 7](#_Toc135038862)

# 实验要求

1.使用华为mindspore框架完成手写数字识别和鸢尾花三分类任务

2.完成kaggle手写数字比赛，其中比赛题目（摘自Kaggle官网）为：

“MNIST（Modified National Institute of Standards and Technology）是计算机视觉领域事实上的“Hello World”数据集。自1999年发布以来，这个经典的手写数字图像数据集一直被用作分类算法基准测试的基础。随着新的机器学习技术的出现，MNIST仍然是研究人员和学习者的可靠资源。在这个竞赛中，你的目标是从数万张手写图像的数据集中正确识别数字。我们为你提供了一系列以教程形式呈现的代码笔记本，涵盖了从回归到神经网络的各种技术。我们鼓励你尝试不同的算法，亲自学习哪种方法效果好，以及不同技术之间的比较。”

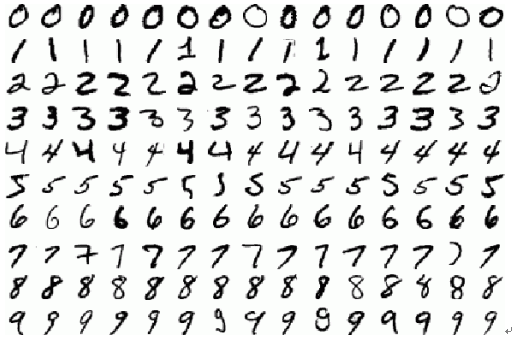
数据集如图1所示，需要我们做出分类任务，判断手写数字实际数字是什么。

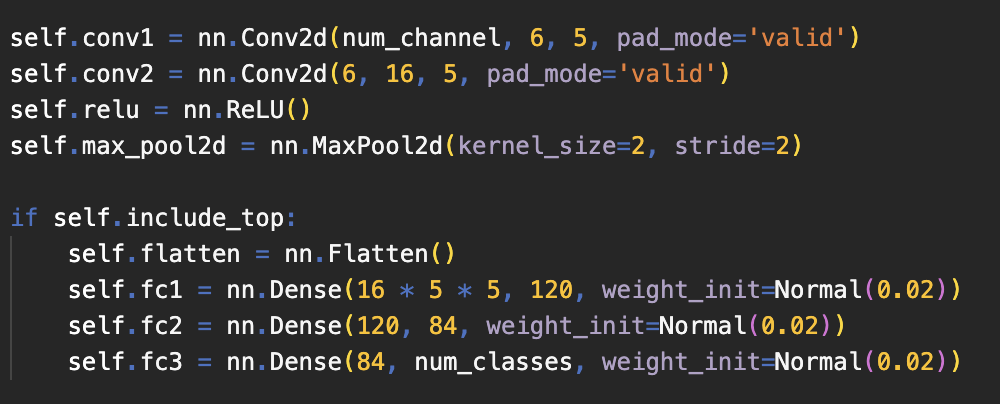
图1 手写数字识别数据集

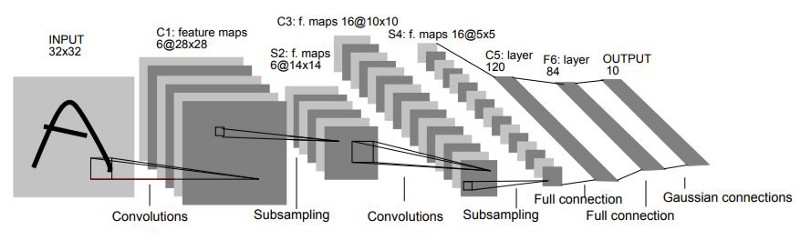
# 算法设计与实现

## 1.华为mindspore手写数字识别

首先要下载数据集（代码略），然后使用create\_dataset函数实现数据集的读取，定义好相关的resize，rescale，batchsize等超参数之后，对数据集进行预处理，下采用了shuffle，batch，repeat等处理方式，可以提高网络的鲁棒性。

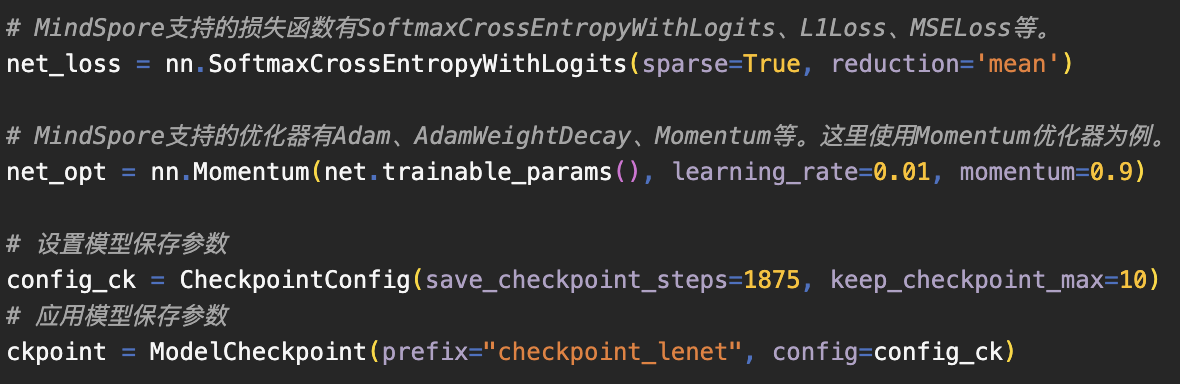
接下来要定义网络结构，我使用的网络结构是lenet-5，对应的网络架构如图





该网络使用了两个卷积层，使用的激活函数为Relu，然后加入了一个maxpool池化层，然后使用include\_top来判断是否需要加入mlp（多层感知机），来进行对应的分类，这大大提高了模型的可迁移性，方便使用预训练的模型，改变最后的mlp层来实现任务（本实验中未使用迁移学习）。

然后是确定损失函数和优化器，设置网络参数保存点（checkpoint）



我选择的损失函数是交叉熵函数，是很经典的机器学习损失函数

优化器的选择是momentum，使用动量的梯度下降算法会在更新权重时引入一个额外的项，该项考虑了之前的权重更新方向，以模拟物体在梯度方向上的滚动（如图2所示）。

具体而言，动量通过维护一个动量变量（通常表示为v），该变量保存了之前权重更新的方向和幅度信息。在每次权重更新时，动量项与当前的梯度相结合，得到新的权重更新方向。这样做的效果是，在平坦区域上，动量可以加速权重更新的速度；而在陡峭的区域，动量可以抑制震荡，使得权重更新更加平滑

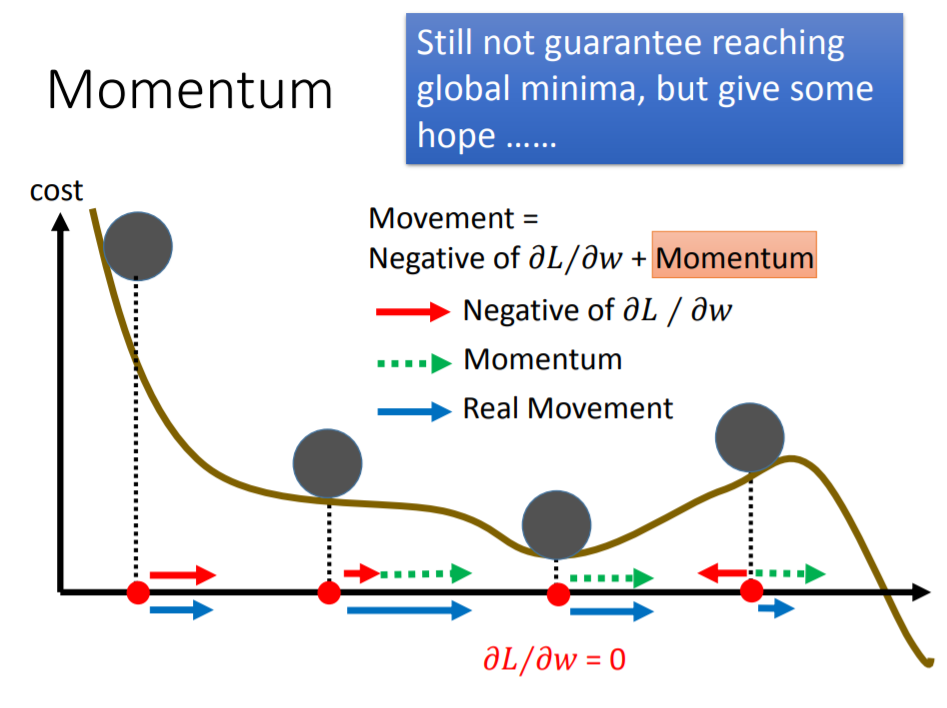


图2 momentum优化器图示

最后是train和test的函数实现

文本

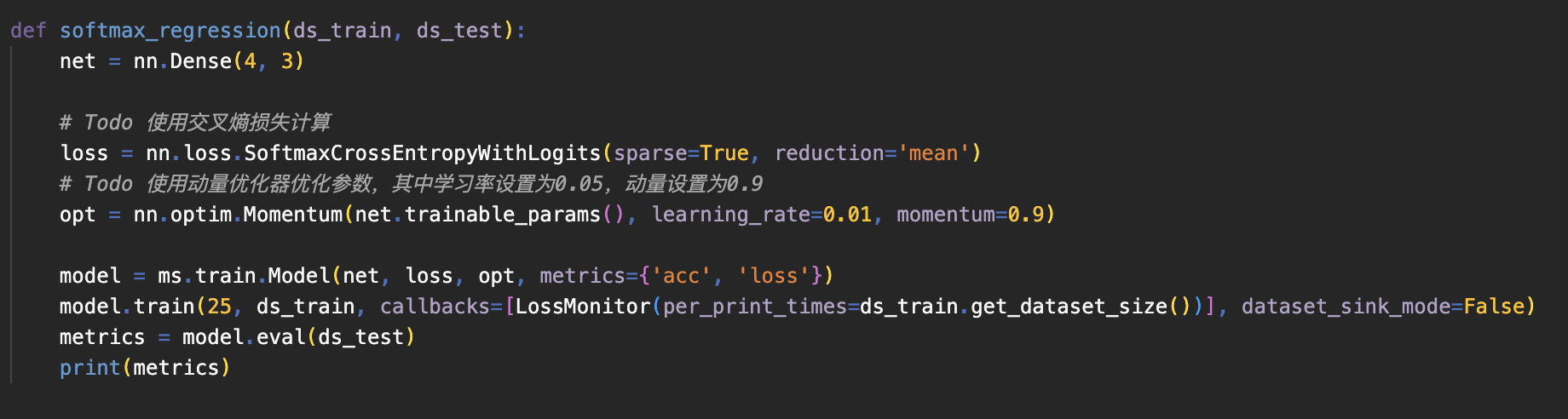
描述已自动生成

调用mindspore框架的train和eval函数，进行训练和验证，最后打印出测试准确率。

## 2.华为mindspore鸢尾花分类问题

首先还是读取数据，将Iris-setosa，Iris-versicolor，Iris-virginica对应为0，1，2三类，然后随机将数据分为8:2两部分，作为训练集和验证集，然后调用mindspore框架中的GenerateDataset生成训练集和验证集。

然后我们定义我们的网络架构，这里我们使用线性回归，使用交叉熵函数和momentum优化器，实验证明他们有较为良好的性能。接下来训练改网络并验证测试集准确率。



## kaggle手写数字分类比赛

首先还是读取数据集，这里的数据集是使用csv表格保存在本地的，表格的数据相当于是手写数字图片做了flatten操作平坦化的结果，我们首先找出对应的训练集和测试集，找到训练数据data和对应的标签lable，然后是用对应的transform变换将他们转回图片。方便送入神经网络

截图里有图片

描述已自动生成

然后我们定义网络架构，由于是要参加kaggle比赛，需要更高的网络精度，相比于实验1，这里使用的网络深度更深，模型更大，能获得更好的性能。文本

描述已自动生成

网络使用了5个卷积层，与实验1不同的是，我加入了dropout层，还有BatchNorm层（用于进行Batch Normalization操作）。

Dropout层是深度学习中一种常用的正则化技术，其作用是在神经网络中随机地将一部分神经元的输出置为零。通过在训练过程中随机丢弃一些神经元，dropout层可以强制网络学习更加鲁棒的特征表示。具体而言，dropout层在每次前向传播时，以一定的概率（通常是0.5）将某些神经元的输出置为零，（如图3所示）从而减少它们的影响。这种随机丢弃的过程类似于集成多个子网络的预测结果，因为每次前向传播时，网络的结构都会有所变化。通过训练多个这样的子网络，dropout层有助于减少网络中神经元之间的依赖关系，增加网络的鲁棒性，提高泛化能力。

Batch Normalization（如图4所示）是一种用于加速深度神经网络训练过程并提高其性能的技术。它通过对网络中每一层的输入进行归一化操作，将特征的分布调整为均值为0，方差为1的标准正态分布。Batch Normalization通过对每个小批量样本的特征进行归一化，缓解了内部协变量偏移的问题。具体而言，对于每个特征维度，Batch Normalization计算该特征在当前小批量样本中的均值和方差，并使用这些统计量对特征进行归一化。然后，通过学习两个可学习参数（缩放因子和偏移量），将归一化后的特征重新缩放和平移，使得网络能够学习适合当前任务的特征表示。其作用是加速网络收敛，增加网络深度

降低网络对初始参数的敏感度等。

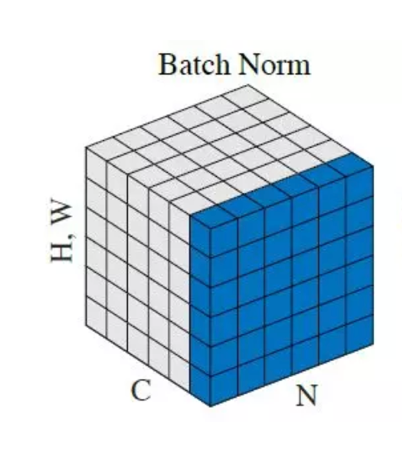
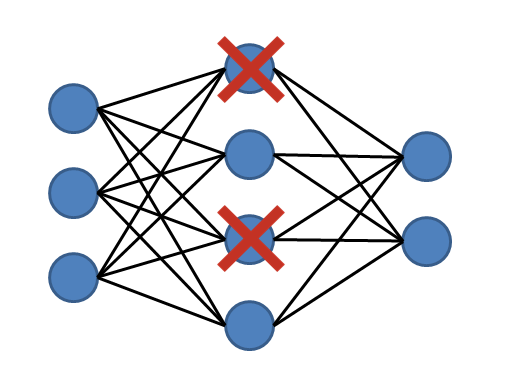


图3 Dropout示意图 图4 Batch Norm示意图

接下来定义一些超参数，损失函数和优化器，我选择训练五次，batchsize设为64，学习率为0.01，momentum为0.9，优化器为SGD，loss使用交叉熵损失。

文本

描述已自动生成

最后就是训练和验证，将预测结果放入一个csv文件来提交。

# 实验环境与平台

表1 实验1硬件环境

|  |  |
| --- | --- |
| 设备 | 规格 |
| 处理器 | Apple Silicon M1 |
| 机带RAM | 16.00 GB |
| 系统类型 | MACOS操作系统, 基于ARM64的处理器 |

表 2 软件环境

|  |  |
| --- | --- |
| 软件 | 版本 |
| Python | 3.9.16 |
| Numpy | 1.23.5 |
| Pandas | 1.4.3 |
| Scikit-learn  Mindspore  Pytorch | 1.1.1  2.0.1  2.0.0 |

# 结果与分析

实验1结果准确率为0.964。

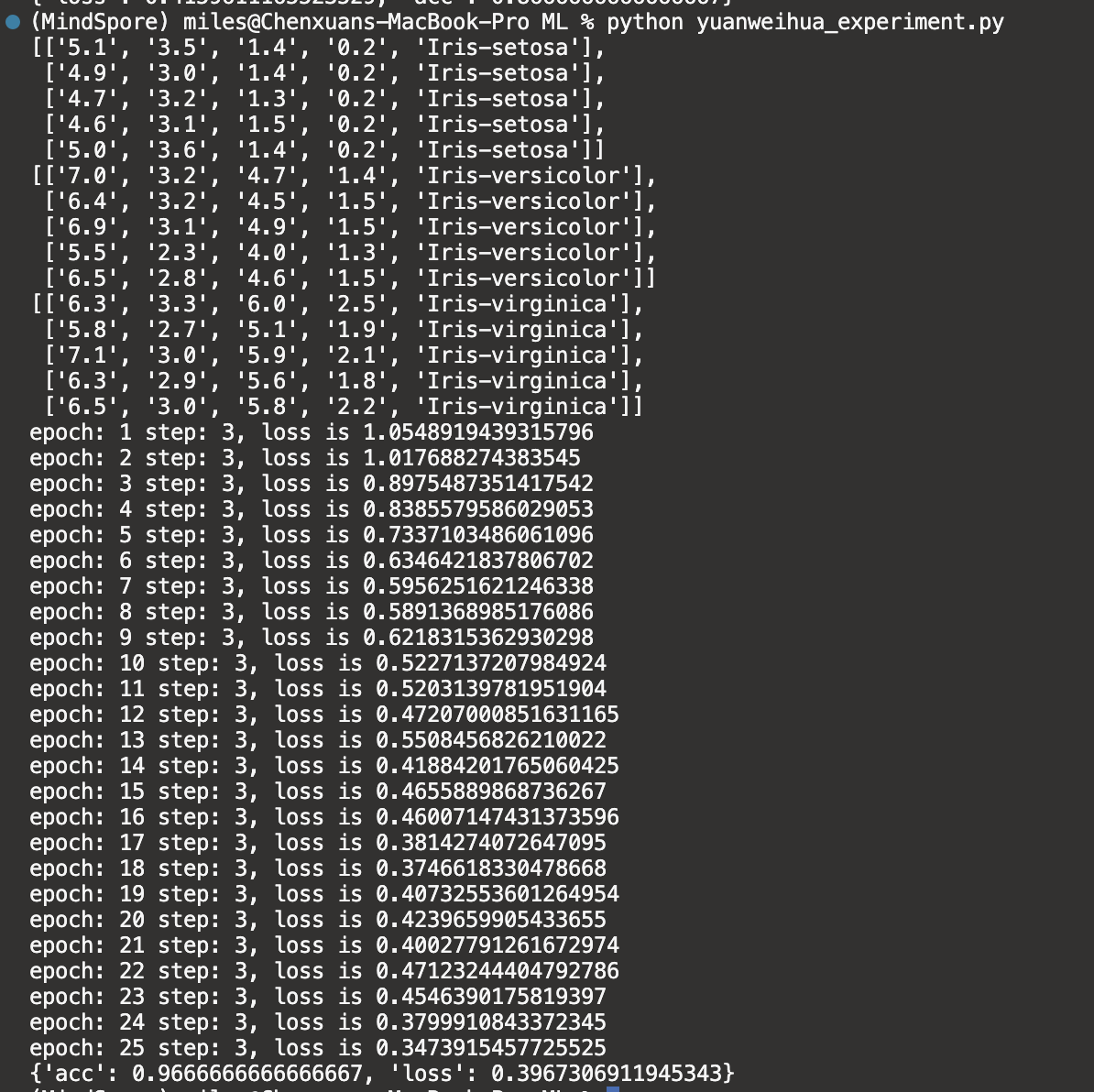
文本

描述已自动生成

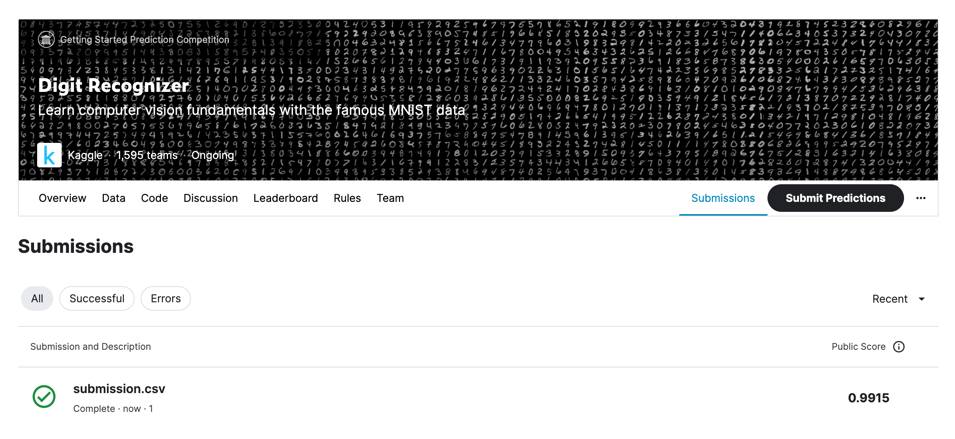
其中的warning是由于华为mindsporce框架很快就不支持原有的transform操作，下面有一个具体的predict测试，输入图片为7，预测结果为7，可见具有较好的分类性能。

实验2结果准确率为0.967。

在训练过程中可以看到损失大致上在渐渐减小，在训练结束时实现了一个较为优秀的分类准确率。但是由于训练样本数据较小，实验的结果受随机的影响较大。



实验3的准确率为0.9915，得益于更大的网络和dropout与batchnorm等更先进技术的加入，准确率明显优于实验1中的结果。



# 5. 个人体会

在本次实验中使用了华为mindspore框架，体验到了国产框架的优秀和便利，国产框架大大加快了我们在开发深度学习网络时候的速度，提高了开发效率。

同时在实验中我也学习了传统的机器学习算法和深度学习算法，利用他们完成了实验目标，体会到了AI和机器学习的强大之处，掌握了相关基本方法。

另外，这次试验也让我对各个模型的特点有了初步印象。在本次实验所选取的各个模型中，两个深度神经架构相似，但是随着深度加深和模型加大，引入新的正则化方法，可以使得准确率大大提高，使我对机器学习和深度学习又了更加深刻的认识。