**mlp-NN课程报告**

学号：20307110315

姓名：周训哲

# 1.任务描述

利用mlp-NN多层感知器神经网络算法实现手写数字的识别与分类。解决了多样化模糊图片的识别分类问题。其中损失函数主要运用了CrossEntropyLoss（交叉熵）和MultiMarginLoss（多分类合页）两种损失函数。

# 2.数据集描述

**采用何种数据集开展实验？**

利用MNIST手写数字集作为学习以及测试的数据集。里面包含了图片集作为识别对象，标签集作为学习判断的标准。

**数据集有什么特点？**

MNIST数据集来自美国国家标准与技术研究所, National Institute of Standards and Technology (NIST)。训练集 (training set) 由来自 250 个不同人手写的数字构成, 其中 50% 是高中学生，50% 来自人口普查局 (the Census Bureau) 的工作人员。测试集(test set) 也是同样比例的手写数字数据。具有极大的测试数据量。

数据集分为训练集和测试集两个部分，其中又细分为图片集和标签集。

图片集主要存储的是图像信息，每一张图片包含28\*28个像素，在数据集中展开转换为一个长度为784的向量数据。在Mnist的训练数据集中mnist.train.images是一个形状为[60000, 784]的张量，第一个维度数字用来索引图片，第二个维度数字用来索引每张图片中的像素点，图片里的某个像素的强度值介于0-1之间。

标签集主要存储的是分类的信息，将每张图片所表示的数字进行存储。Mnist数据集中所有的标签mnist.train.labels是一个[60000, 10]的数字矩阵。在训练集中主要用来学习，以及kNN算法中的分类（利用距离）。在测试集中主要用来对比计算算法的识别准确率。

**数据集的各项统计量是多少？**

训练集中的数据量有60000个，测试集中数据量有10000个。

# 3.方法介绍

## 3.1数据预处理（如有）

**从原始数据到输入模型的数据之间，经过了哪些处理？这些处理有何作用？**

首先，原始数据为ubyte二进制文件，需要对二进制数据进行处理。需要利用open（）函数打开，其中参数为 ‘rb’，其含义为将文件以二进制文件形式打开。

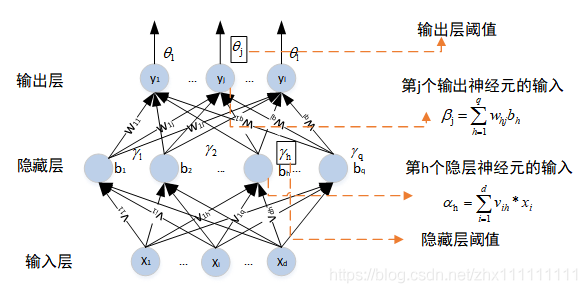
对于图像数据（image）是一个3维的数据集（60000\*28\*28）。开头为信息标识，有四个变量，魔数（magic）为2051，数据集大小（size）为60000，图像的行、列都是28，均通过unpack（）函数读取。然后将图像数据从28\*28的二维数据进行转换为长度为748的一维向量images。

对于标签数据（label）是一个一维的数据集（60000），开头也是信息标识，有两个变量，魔数（magic）为2049，数据集大小（size）为60000，也是通过unpack（）函数读取，最后利用append（）函数，将标签存储在向量labels中。

## 3.2算法描述

**核心思想：**

多层感知机就是含有至少一个隐藏层的由全连接层组成的神经网络，且每个隐藏层的输出通过激活函数进行变换。隐藏层与输入层是全连接的，假设输入层用向量X表示，则隐藏层的输出就是 f (W1X+b1)，W1是权重（也叫连接系数），b1是偏置，函数f可以是常用的sigmoid 函数或者tanh函数或者relu函数（本次实验为了节省计算时间，采用了最简单的relu函数），输出函数采用适合非线性多分类的softmax函数。



由于多层感知机算法的局限性，提出了误差逆传播算法即BP算法。BP算法基于梯度下降策略，以目标的负梯度的方向对参数调整，给定学习率，控制着算法每一轮迭代中更新步长，若太大则容易振荡；若太小则收敛速度太慢）。

在BP算法训练的过程：

**1将输入示例提供给输入神经元；**

**2对输入数据进行计算，并逐层向前传播，产生输出结果；**

**3计算输出结果的误差；**

**4将误差逆向传播至隐层神经元；**

**5根据误差，调整参数（包括连接权和阈值），从2开始反复迭代，直到满足条件。**

**整体流程：**

总体来说，kNN分类算法包括以下4个步骤：

①准备数据，对数据进行预处理。

②定义MLP类，设置四个感知机网络，激活函数为relu，输出函数使用softmax

③训练神经网络，使用bp算法，不断迭代得到最优参数（总共迭代100次）

④对测试集进行回测，计算准确率

**缺点：**

（1）有隐藏层的MLP包含一个非凸性损失函数，存在超过一个最小值，所以不同的随机初始权重可能导致不同的验证精确度；

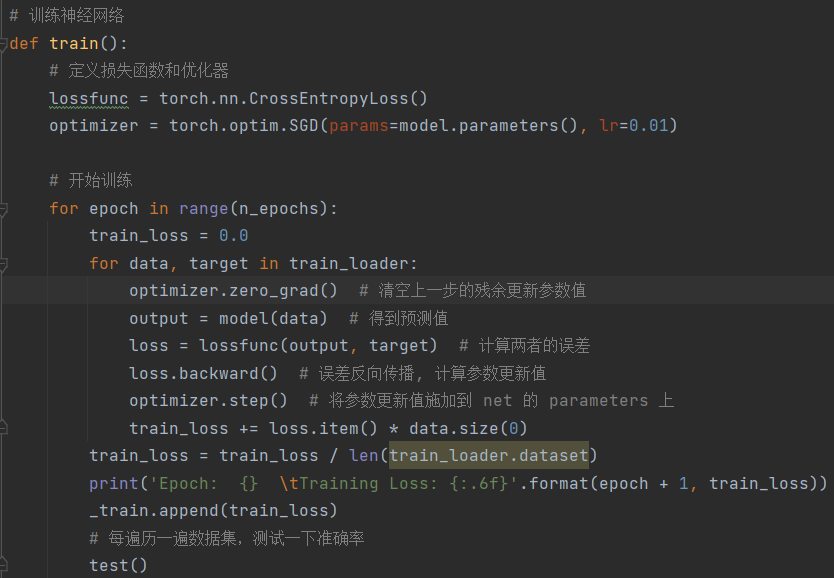
（2）MLP要求调整一系列超参数，如隐藏神经元、隐藏层的个数以及迭代的次数；

（3）MLP对特征缩放比较敏感。

**训练和预测过程:（具体实现思路）**

训练过程即对mlp使用bp算法进行学习的过程。

具体的代码实现如下，每训练一次都进行一次参数的更新和准确率的输出。



预测的过程则是将参数代入测试集，进行神经网络的顺向传输，判断符合预期label的占比并统计

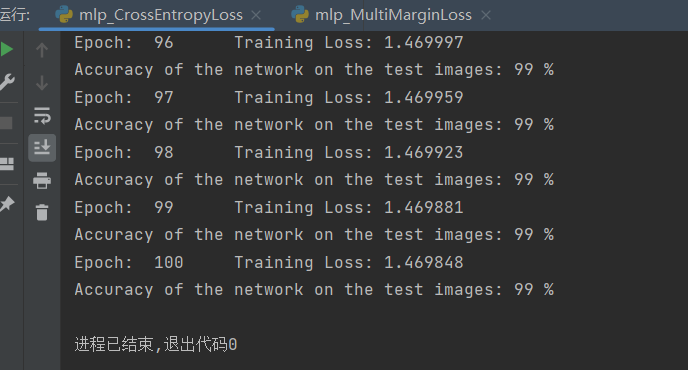


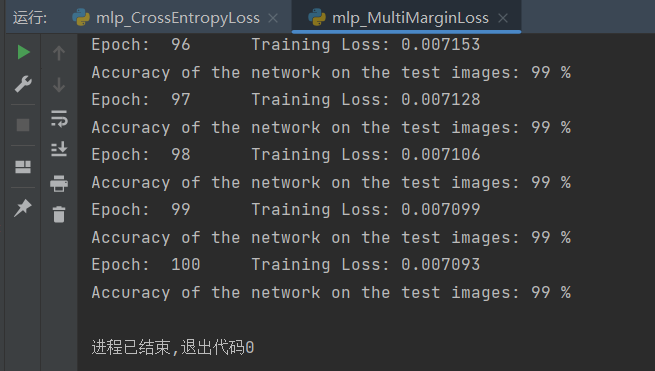
# 4.实验结果分析

## 4.1评价指标

该算法训练集学习之后，对测试集样例的图像文件进行识别与预测，将预测结果与标签集比对，最后计算得到10000个测试数据的准确率。

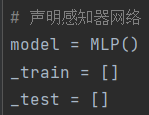
## 4.2定量评价结果



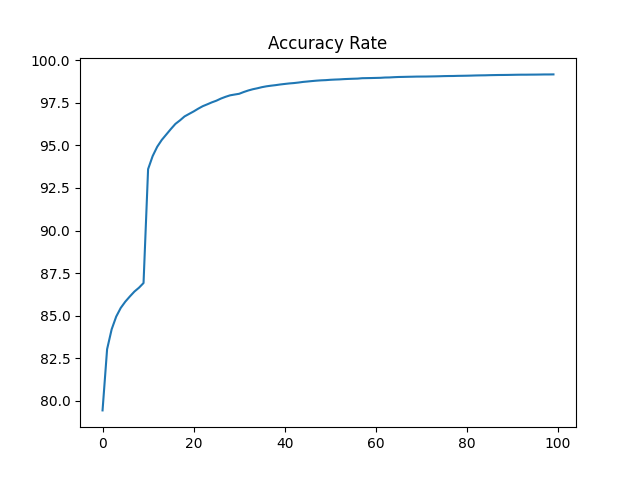
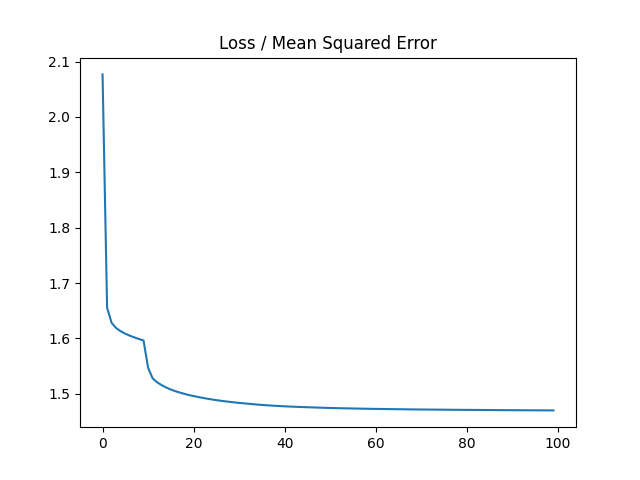


比较两种损失函数的损失以及其准确率，发现最终学习的结果准确率最高都可以达到99%（之所以强调最高，是因为在batch\_size的设置上有所不同，经过多次测试，发现最适合交叉熵的值为8，最适合多分类合页的值为4，其参数测试过程由于每次运行时间较长，只测试了20左右的epoch（循环），故不作截图参考比较）。除了准确率，很明显可以发现交叉熵的均方误差损失是远大于多分类合页的，所以在多分类数字集实验中采用多分类合页的损失函数得到的误差更小。

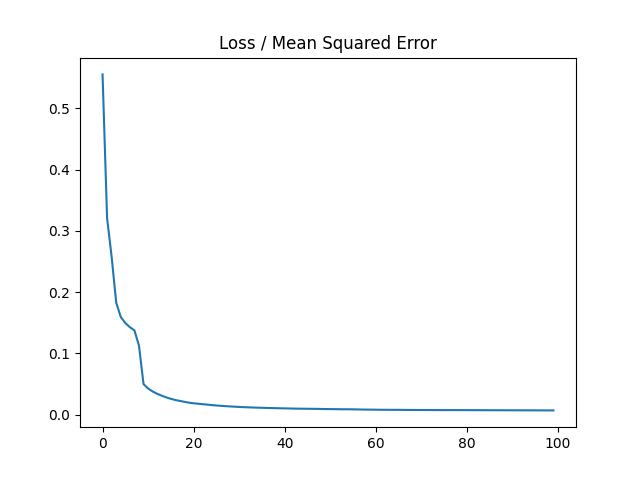
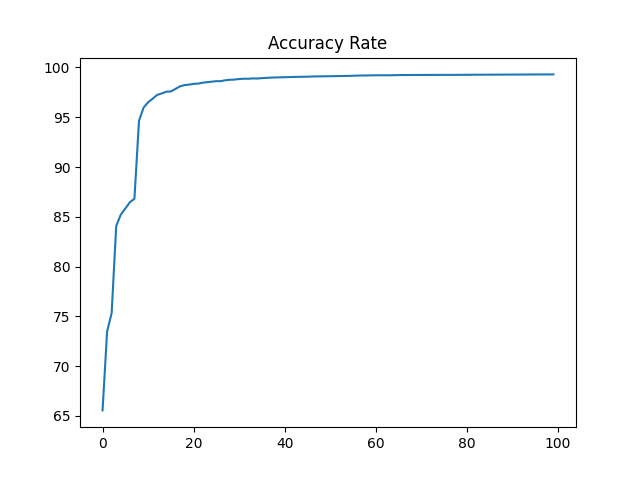
## 4.3可视化结果

利用list存储每次的准确率和均方误差，最后在图表中进行表示。

**交叉熵：**



**多分类合页：**

可以看见，随着迭代次数的增加，准确率不断上升，误差损失不断下降，说明多层感知机学习有明显效果。

# 总结

本次实验主要学习了最简单的神经网络算法——多层感知机，在本次实验中，我学会了神经网络算法中的bp算法的具体工作原理以及如何使用代码实现。除此以外，我了解了多层感知机的不同损失函数以及激活函数所带来的结果差异。

本次实验主要采用relu作为激活函数，一个原因是因为为了计算方便，节约时间，采用最简单的激活函数可以提高算法效率，另一方面，采用relu函数作为线性的分类函数，对于简单的分类来说，可能效果还要比非线性的分类函数效果更好（具体参考svm中的sigmoid）。同时采用softmax作为输出函数的原因也是因为，对于非线性多分类模型来说，softmax这种非线性模型效果可能更好。