1引言

1.1编写目的

详细设计是设计的第二个阶段，这个阶段的主要任务是在概要设计书基础上，对概要设计中产生的功能模块进行过程描述，设计功能模块的内部细节，包括算法和详细数据结构，为编写源代码提供必要的说明。 概要设计解决了软件系统总体结构设计的问题，包括整个软件系统的结构、模块划分、模块功能和模块间的联系等。详细设计则要解决如何实现各个模块的内部功能，即模块设计。具体的说，模块设计就是要为已经产生的图书管理各子系统设计详细的算法。但这并不等同于系统实现阶段用具体的语言编码，它只是对实现细节作精确的描述，这样编码阶段就可以将详细设计中对功能实现的描述，直接翻译、转化为用某种程序设计语言书写的程序

1.2项目背景

神经网络的训练对于初学者而言比较困难，容易找不到训练的方向。该项目可将训练过程和结果直观表示出来，让初学者初步了解到各个参数对于训练的影响。

任务提出者：学校课程设计

开发者：吴浩林 麦善锐 胡婧媛

用户：神经网络入门学习人员

1.3定义：

learning Rate:学习率。学习率决定着目标函数能否收敛到局部最小值以及何时收敛到最小值。

batchsize:批大小。在深度学习中，每次训练在训练集中取batchsize个样本训练。

epoch:迭代次数。1个epoch等于使用训练集中的全部样本训练一次

train data size:训练集数据大小

test data size:测试集数据大小

cnn:卷积神经网络

mlp:多层感知器

1.4参考资料

[https://tensorflow.google.cn/js](https://tensorflow.google.cn/js：tensorflow.js)：官方网站，包含教程和api文档

<https://github.com/tensorflow/tfjs-examples>：官方githup上的实例存储库

<https://www.papaparse.com/>：一种csv文件的解析器

<https://d3js.org/>：一种JavaScript的数据操作库

2总体设计

2.1需求规定

2.1.1文件上传功能，接收用户的训练数据

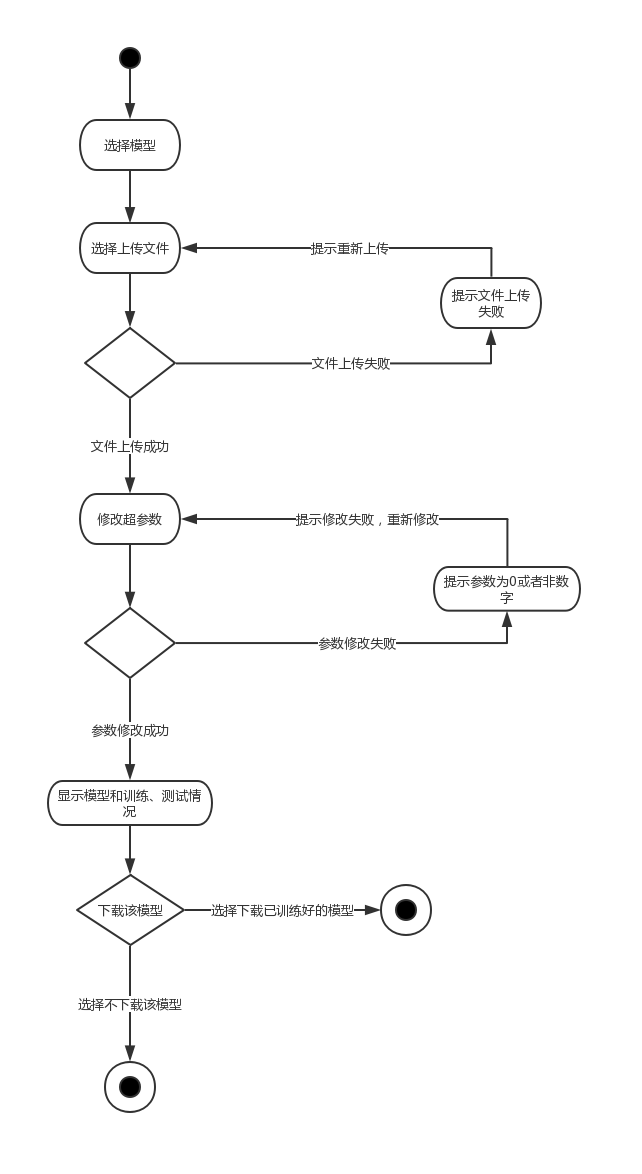
2.1.2参数修改功能，修改训练模型的超参数

2.1.3下载模型功能，用户可以下载训练好的模型

2.2软件结构

3模型设计说明

3.1功能流程图



3.2mlp模型描述

3.2.1结构

5个网络层，两个为输入输出层，输入和输出都是一个节点。三个隐藏层都是1024个节点，后两个隐藏层使用了relu作为激活函数

3.2.2性能

该模型因为结构比较简单，训练速度较快，对于二维数据拟合效果很好

3.2.3输入项

一个两列的csv文件，第一行为列为，训练数据的行名为x，标签值的行名为y，数据类型为float。

3.2.4输出项

输出为一个预测值，数据类型为float

3.2.5算法

优化算法：随机梯度下降算法

损失函数：均方误差

3.2.6接口

使用了d3.js的外部接口来读取数据。对外接口为mlp\_run()

3.3cnn模型描述

3.3.1结构

5个网络层。分别是一个卷积层，一个最大池化层，一个卷积层，一个最大池化层和一个输出全连接层。第一个卷积层8个卷积核，卷积核大小为5\*5，步长为1，激活函数为relu。最大池化层均为，一次池化大小为2\*2，步长为2\*2。第二个卷积层卷积核为16，卷积核大小为5\*5，激活函数为relu。最后一个全连接层10个输出节点。

3.2.2性能

该模型拟合图像效果较好，但需要较长的训练时间和较大的训练集

3.2.3输入项

四个csv文件，分别是训练集数据和标签，测试集的数据和标签。均不需要列名。数据集的每一行为一个图像平铺后的像素值，标签的每一行为对应数据集的标签。

3.2.4输出项

输出为10种标签的概率，数据类型为float

3.2.5算法

优化算法：随机梯度下降算法

损失函数：交叉熵

3.2.6接口

使用了read\_data.js的外部接口来读取数据。对外接口为cnn\_run()