PJ1文档 简易分布式神经网络

本次pj要求实现一个简易的分布式神经网络根据学生数据对学生进行分类。

1. 整体架构
2. 代码包目录结构

pj1

|--client.py client进程

|--master.py master进程

|--ws.py worker server进程

|--ps.py parameters server 进程

|--network\_ps.py 包含Network类，和一些ps进程需要调用的函数

|--generate\_data.py 生成训练和测试数据的脚本

|--student\_data.csv pj中预先给出的数据集

|--train\_1.csv student\_data.csv的前300行数据，可以用作训练数据

|--test\_1.csv student\_data.csv的后100行数据，可以用作测试数据

|--test.csv 脚本生成的测试集数据

|--train.csv 脚本生成的训练集数据

2、项目简介

本次项目使用四个进程模拟分布式系统，分别是client, master, ps, ws，为了增加分布式系统的仿真性，各个进程之间使用socket进行通信。运行时，系统首先开始进行分布式的训练，系统使用过反向传播算法的bp神经网络结构，神经网络的结构可在client中进行灵活调整，支持多层隐含层，支持每层神经元数量调节，默认神经网络结构为输入层、输出层加单层隐含层，隐含层的神经元数量为10。经过100次的训练后，用户可以进行输入，用户分别输入gre, gpa, rank三项数据后，数据交至master, master对数据进行预处理normalization后，将数据交由parameters server, 由worker server进行计算，计算时数据值来自parameters server, 得到计算值后，将返回至client, client接收到结果后进行输出，然后等待下一次用户输入。

在前期的训练过程中，神经网络的结构参数，训练集，测试集在client中，由client经socket发送至master，master接收到相关数据初始化数据集数据，并进行预处理normalization，然后数据传至ps中，ps根据结构参数初始化网络结构，接下来ps负责存储网络参数并在需要时将参数提供给ws，同时负责训练过程当中的反向传播计算，ws进行正向传播计算，二者协作完成训练任务。训练结束后，将发送信号通知master，client，等待用户的下一步输入。

二、函数及类的功能介绍

1、类的介绍

本次项目中为了进一步仿真分布式系统，为了能够尽量做到将系统中的不同部署到不同服务器上就可以进行真正的分布式计算，特地使用了四个独立的进程，进程间使用socket通信，所以系统主体是四个独立进程，没有过多定义类，在项目中只使用了一个类，即Network类，用于在parameters server中存储神经网络的参数。对于Network的具体介绍如下：

|  |
| --- |
| class Network |
| attributes:  para['learning\_rate'] 存储神经网络训练时的学习率  para['size'] 存储神经网络的结构参数，为list类型，默认为[3,10,2]  para['hl\_num'] 神经网络隐含层层数  para['inputs']  para['expec\_outputs']  para['outputs']  para['outputs\_bias']  para['outputs\_weight']  para['outputs\_in']  para['hl\_inputs']  para['hl\_outputs']  para['weights']  para['biases']  para['thf\_deris']  para['hl\_out\_deris']  para['weight\_deris']  para['thf\_out\_products']  para['bias\_deris']  para['outputs\_deris']  para['outputs\_weight\_deris']  para['outputs\_bias\_deris']  para['outputs\_in\_deris']  para['outputs\_products'] |
|  |