

Report



Deep Learning group

LiNa

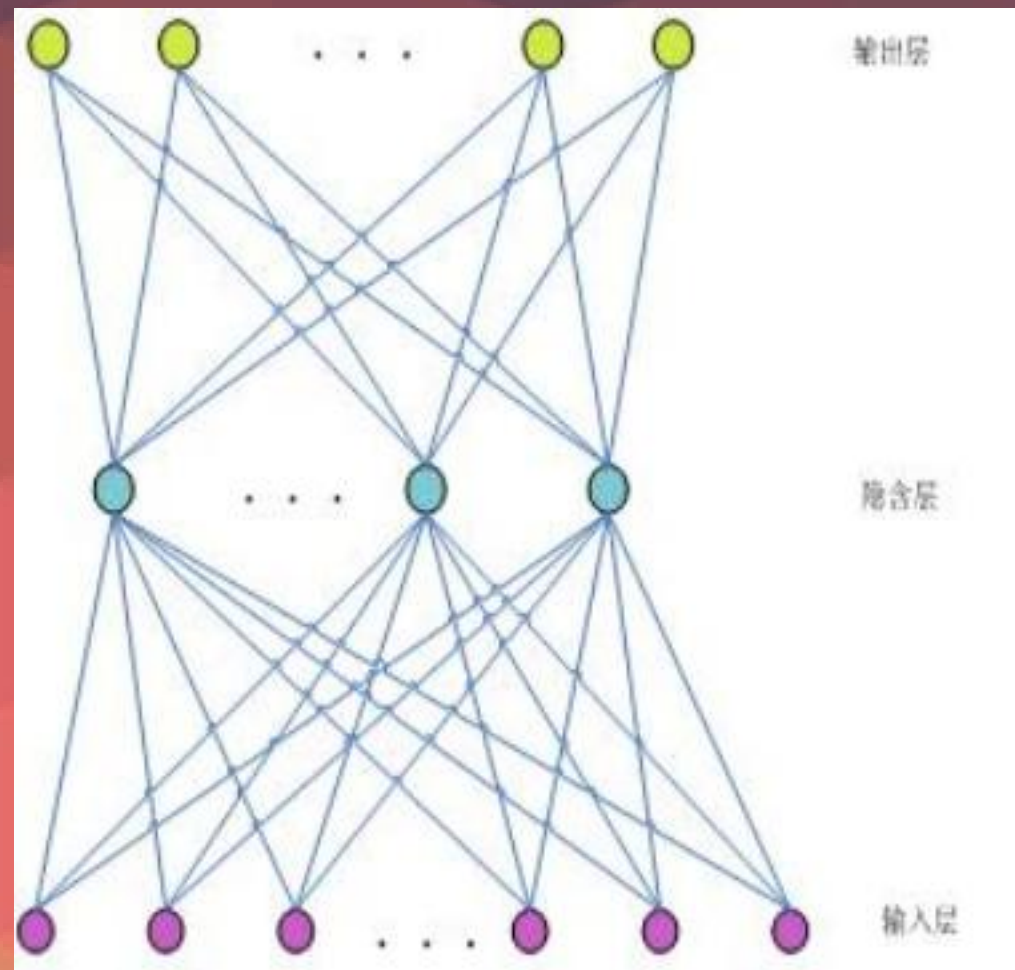
September 22. 2016

CONTENTS

- 1 初识MLP和BP算法
- 2 基础程序编写
- 3 思考反思

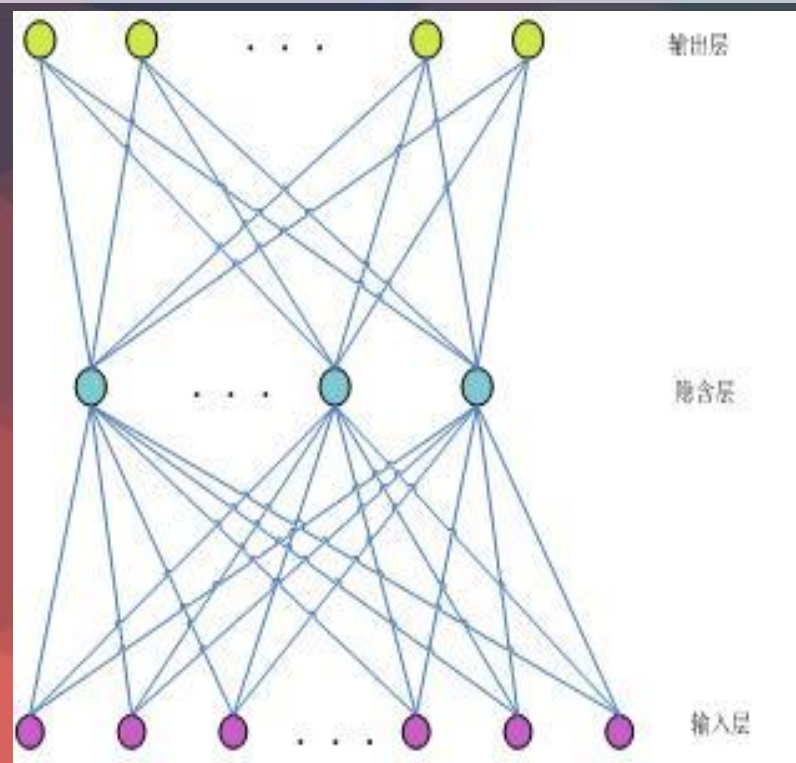
1.初识MLP多层感知器 (Multi-layer Perceptron)

- 1.前向结构的人工神经网络，映射一组输入向量到一组输出向量。
- 2.有向图，多个节点层组成，每层全连接。
- 3.MLP的参数是各个层之间的连接权重 w 以及偏置 b ，求解最佳的参数是个最优化问题
- 4.梯度下降法让权重的值调整到最佳（反馈）
- 5.隐藏结点最好比输入结点的维度大。



1.初识MLP多层感知器 (Multi-layer Perceptron)

- 过程：
- a.前馈forward：
- 已知输入 u ，权值 w （随机赋值），偏置结点 b ，利用激发函数 $f(u)$ ，进而输出 $y=f(aw+b)$
- 每层单元全连接加权传递
- b.反馈backward：
- 已知输出 y ，输出层误差函数 E ，通过对权值的求导，不停迭代，求各层令误差最小的最优权值 w
- $w = w - \alpha (dE/dw)$



1.初识MLP多层感知器 (Multi-layer Perceptron)

- BP算法（误差反向传播 Error Back Propagation）
- 目的：
- 通过梯度下降、链式法则的方法，找到一组能最大限度减小误差的权重
- 作用：
- 反向传播算法的监督学习方法常被用来训练MLP，更新权值

3.基础程序编写

- 1.文件读取

- 打开文档进行读取,
- 分析给出数据形式

(用户ID, 标签, 参数1, 参数2,, 参数30)

- 对每行执行for循环遍历
- 标签进行1/0赋值表示有无疾病
- 关闭文档

注: 数据来源

uci dataset (Breast Cancer dataset)

569组, 每组数据30个参数值

```
#include<iostream>
#include<stdio.h>
using namespace
int main(){
    FILE *fp;
    fp=fopen("uci","r");
    int a[569];
    char tmp;
    long int m;
    char label;
    double data[569][30];
    for(int i=0;i<569;i++){
        fscanf(fp,"%ld",&m);
        fscanf(fp,"%c",&tmp);
        fscanf(fp,"%c",&label);
        if(label=='M')
            a[i]=1;
        else
            a[i]=0;
        fscanf(fp,"%c",&tmp);
        for(int j=0;j<29;j++){
            fscanf(fp,"%lf",&data[i][j]);
            fscanf(fp,"%c",&tmp);
        }
        fscanf(fp,"%lf",&data[i][29]);
    }
    fclose(fp);
```

3.基础程序编写

2.文档数据写入，实现数据分离

- 新建两个文档
- 采用for循环遍历
- 分别读取标签各列的每行数据、标签
- 写入相应文档实现数据分离

```
FILE *raw;  
raw=fopen("data","w");  
for(int i=0;i<569;i++){  
    for(int j=0;j<30;j++){  
        fprintf(fp,"%lf\t",data[i][j]);  
    }  
    fprintf(fp,"\n");  
}  
fclose(raw);  
FILE *la;  
la=fopen("lab","w");  
for(int i=0;i<569;i++){  
    fprintf(la,"%d\n",a[i]);  
}  
fclose(la);  
return 0;  
}
```


3.基础程序编写

- 问题出现：此数据集行列太多
- 行数读取
- 每行使用getline函数读取
- getline(读取文件的流，读取内容存放位置，行结束的标志)

```
#include <iostream>
#include <string>
#include <fstream>
using namespace std;
int main(){
    ifstream fileStream;
    string tmp;
    int count=0;
    fileStream.open("/home/lina/programme/uci");
    if(fileStream){
        while(getline(fileStream,tmp,'\n')){
            count++;
        }
        fileStream.close();
    }
    cout<<count<<endl;
    return 0 ;
}
```


3.基础程序编写

- 问题出现：此数据集行列太多
- 列数读取
- 由于uci文件结构
- 列数读取采用：每行读取完毕回车结束，读取有多少个逗号，进行相应计算（由于数据结构最后一列没有逗号，计算count需要-1）

```
#include <iostream>
#include <stdio.h>
#include <string>
#include <fstream>
using namespace std;
int main(){
    ifstream fin("/home/lina/programme/uci");
    string str;
    getline(fin,str);
    int count=0;
    for(int i=0;str[i]!='\0';i++){ //到一行为止，每一行的逗号个数
        if(str[i]==',')
            count++;
    }
    cout<<count-1<<endl;
    return 0;
}
```


3.基础程序编写

- 3.数据集归一化
- 先打开文档，分析表格结构，对每行每列进行读取，以便后面读出使用

```
#include<iostream>
#include<stdio.h>
using namespace std;
int main(){
    FILE *fp;
    fp=fopen("/home/lina/programme/data","r");
    double data[569][30];
    for(int i=0;i<569;i++){
        for(int j=0;j<30;j++){
            fscanf(fp,"%lf",&data[i][j]);
        }
    }
    fclose(fp);
    double max[30];
    double min[30];
    for(int i=0;i<30;i++){
        max[i]=-9999;
        min[i]=9999;
    }
    double data2[30][569];
    for(int i=0;i<30;i++){
        for(int j=0;j<569;j++){
            data2[i][j]=data[j][i];
        }
    }
}
```

3.基础程序编写

3.读取每“列”的min,max
采用 $(x-\min)/(\max-\min)$
归一化公式，最后进行
输出

注：每“列”min,max读
取时，对数据进行转置
后，一行为一系列，循
环遍历确定行，再确定
列数，取min,max值

```
}  
for(int i=0;i<30;i++){  
    for(int j=0;j<569;j++){  
  
        if(max[i]<data2[i][j]){  
            max[i]=data2[i][j];  
        }  
        if(min[i]>data2[i][j]){  
            min[i]=data2[i][j];  
        }  
    }  
}  
for(int i=0;i<30;i++){  
    for(int j=0;j<569;j++){  
        data2[i][j]=(data2[i][j]-min[i])/(max[i]-min[i]);  
    }  
}  
FILE *newfile;  
newfile=fopen("/home/lina/programme/result","w");  
for(int i=0;i<569;i++){  
    for(int j=0;j<30;j++){  
        data[i][j]=data2[j][i];  
        fprintf(newfile,"%f\t",data[i][j]);  
    }  
    fprintf(newfile,"\n");  
}  
fclose(newfile);  
return 0;
```


3.基础程序编写

- 针对MLP思路
- a.读数据
 - 定义变量、数组、指针
 - 数据初始化（高斯函数等）
- b.前馈
 - 一维数组bias
 - 二维数组数据
- c.反馈
 - 偏导
- 数据来源： uci dataset
- Breast Cancer dataset
- 569组， 每组数据30个参数值，给出有无疾病结果

4.反思总结

- 1.ubuntu系统分区时根区要在最后分，否则其他区无法分开
- 2.登陆外网，搜索hosts，下载后粘贴在/etc/hosts里面，地址在根目录，需要输sudo gedit /etc/hosts才能编辑。
- 3.QT软件读取文件要写全文件路径

Q：不知道从何下手编起

A：从老师推荐视频看起，多动手实践，从网上看一下代码，尽快写出MLP。

谢谢大家

