

# 结果说明

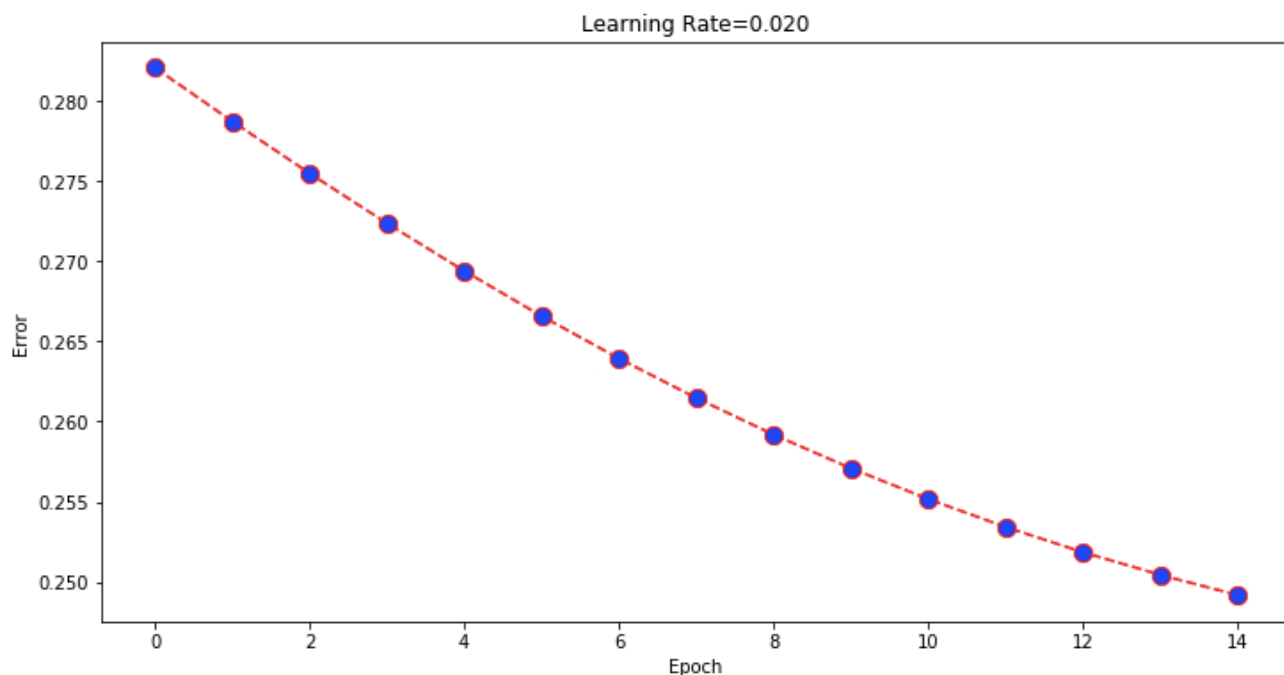
汪楚文 2018202114 05/05/2020

Copyright © 2020- by Wangchuwen. All rights reserved.

## 损失收敛图

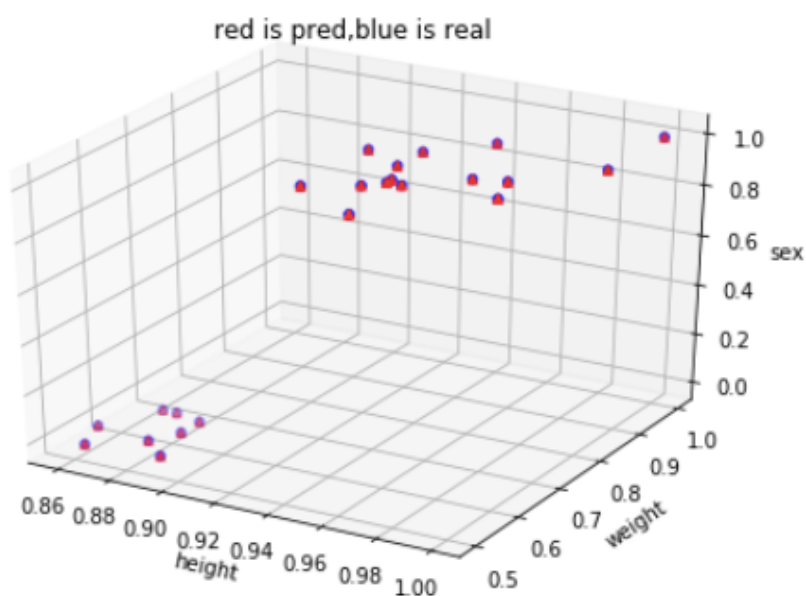
将奇数同学做训练集，偶数同学做测试集。在不同的learn\_rate 以及epochs下，测试的loss值有所不同。当epochs过小时，拟合程度过小，loss较大；当epochs过大时，容易出现过拟合的现象，loss值相对有所上升。

当epoch=15, learn\_rate=0.020时，损失收敛图如下：



## loss值

通过scatter3D用散点标出预测值和真实值，得到的结果如下：



**正确率：100%**

预测值为：

[0. 1. 0. 1. 1. 1. 1. 0. 0. 0. 0. 1. 1. 0. 1. 1. 1. 1. 0. 1. 1. 1. 1. 0.]

真实值为：

[0. 1. 0. 1. 1. 1. 1. 0. 0. 0. 0. 1. 1. 0. 1. 1. 1. 1. 0. 1. 1. 1. 1. 0.]

经过反复调试参数，取epoch=10，learn\_rate=0.020时可以得到测试集上100%的预测准确率。红色三角形为预测值，蓝色为真实值，两者完全重合。

---

## 数据预处理

在进行数据预处理的时候，根据标签生成0，1.

```
def judge(x):  
    if '男' == x:  
        return 1  
    elif '女' == x:  
        return 0  
dataframe['MW'] = dataframe.S.apply(lambda x: judge(x))
```

将身高体重进行归一化处理

```
X_train= normalize(X_train, axis=0, norm='max')
```

```
X_test= normalize(X_test, axis=0, norm='max')
```

在经过多次试验后，发现把normalize的norm参数设置为max比设置为l1或l2要好，故本实验对身高体重的归一化处理为：

$$x^* = \frac{x - \min}{\max - \min}$$

## 神经网络初始权值与bias设置

# 权重，Weights

```
self.w1 = 0.5734666654181668  
self.w2 = -0.7347844458644603  
self.w3 = 0.35871598754027156  
self.w4 = 0.17227684753124822  
self.w5 = -0.05344509263243259  
self.w6 = 1.3818657559335772
```

# 截距项，Biases

```
self.b1 = -0.013258754892991331  
self.b2 = -0.8288385779821897  
self.b3 = -0.5300464985969835
```

经实验，将初始值如上测试代替np.random.constant，此神经网络的稳定性会更好  
经设置，此神经网络在此测试集的预测准确率已稳定在100%

---

## d\_L\_d\_ypred的选取

由于男女的预测结果是离散值，故把神经网络的某些变量进行了round()处理，如：  
 $d\_L\_d\_ypred = -2 * (y\_true - \text{round}(y\_pred))$ 与 $d\_L\_d\_ypred = -2 * (y\_true - y\_pred)$ 相比，前者的反向传播效果更好。

## 目录结构

```
NNpython/  
|  
|-- NNpython.ipynb  
|-- README.pdf  
|-- classdata.csv
```

[超链接!\[\]\(e78f798d4ea5c530c9db49e7d26e6b95\_img.jpg\)： 点击此处可在github中查看文件](#)