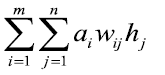
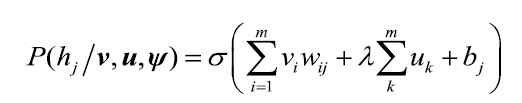
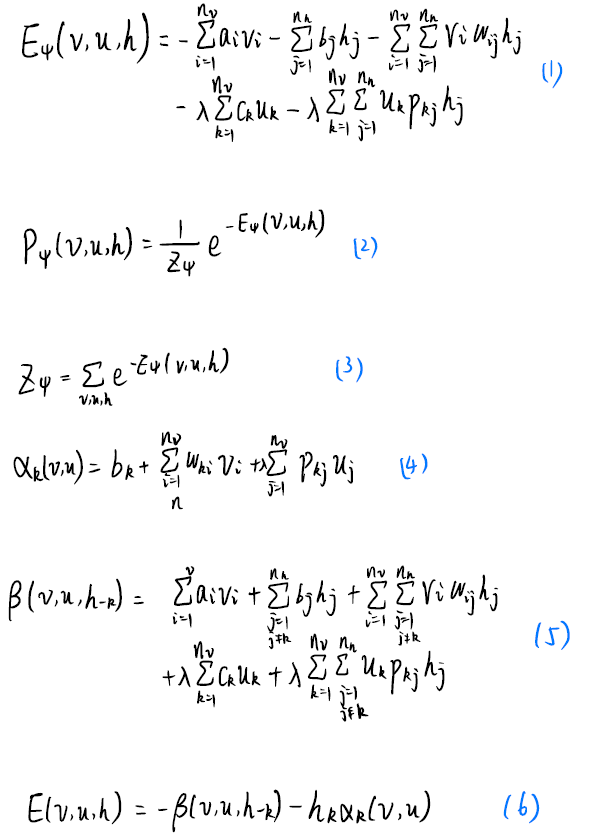
# 推导

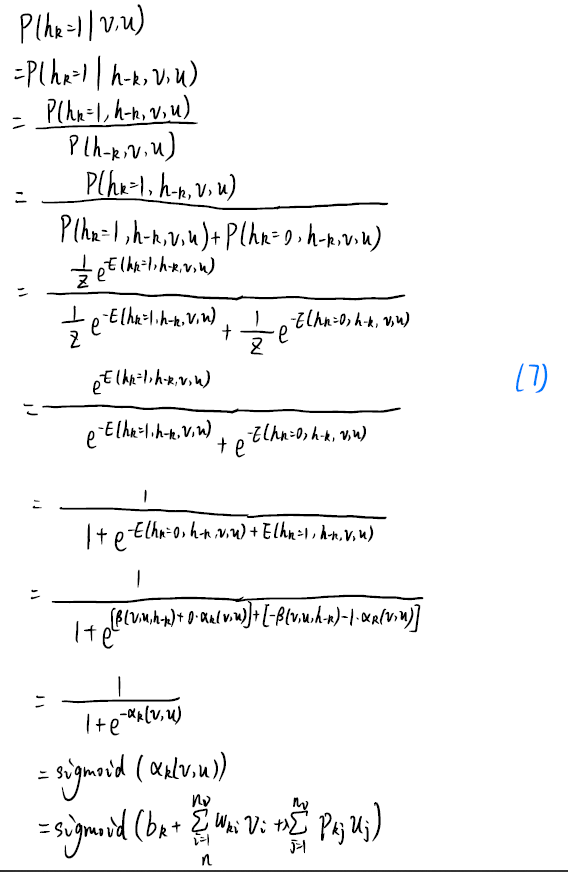


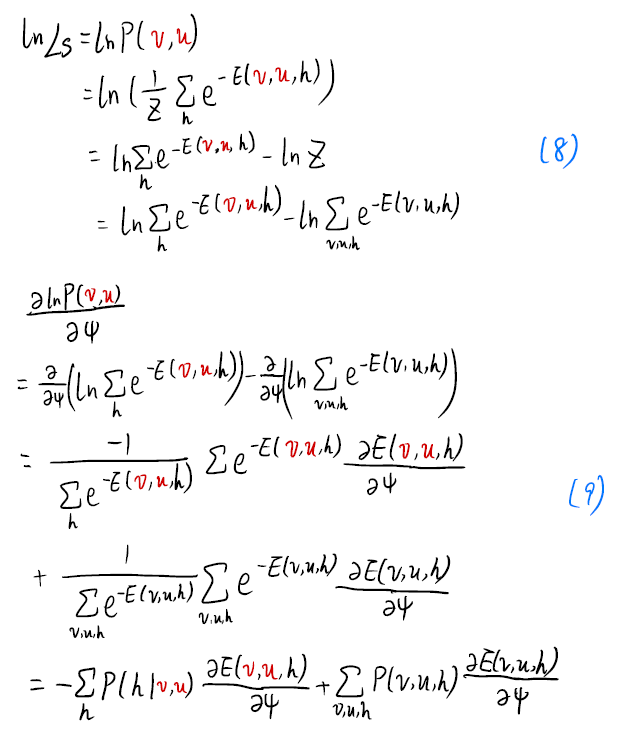
感觉这个ai是vi

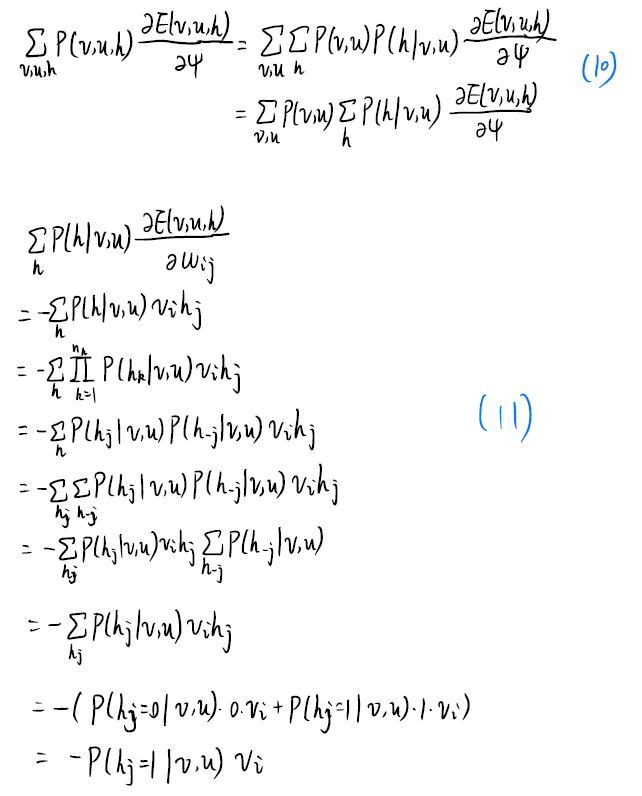


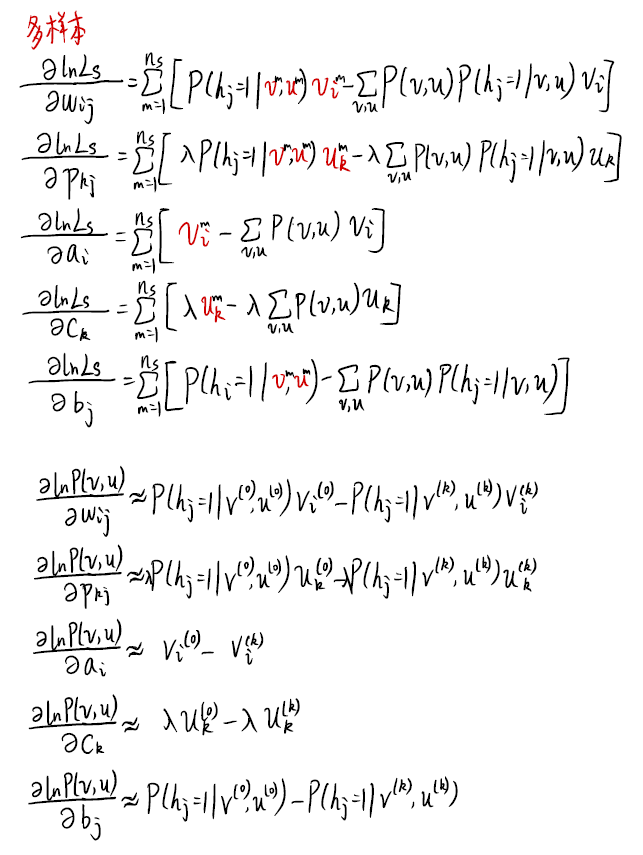
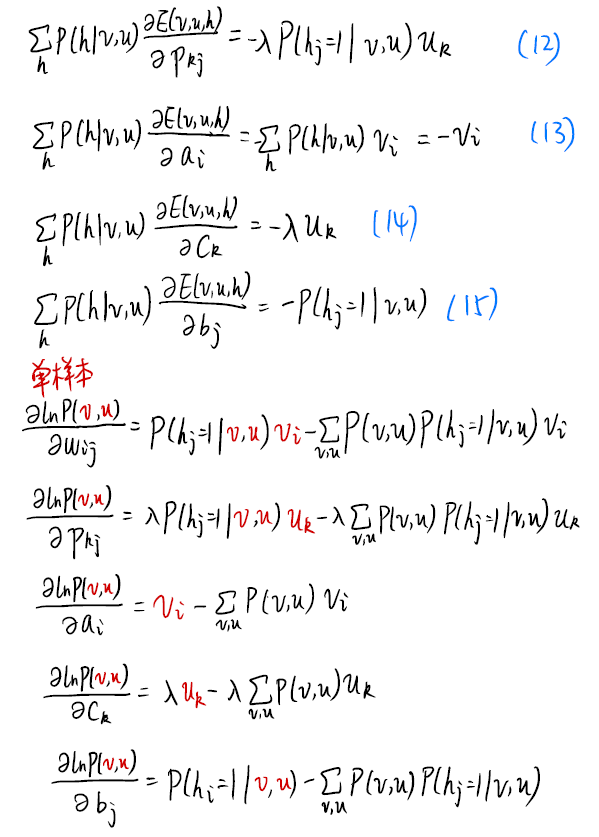
这个推导出来与（7）不一致











# 奇异值分解

SVD可以理解为：将一个比较复杂的矩阵用更小更简单的3个子矩阵的相乘来表示，这3个小矩阵描述了大矩阵重要的特性。

可用np.linalg.svd(input)

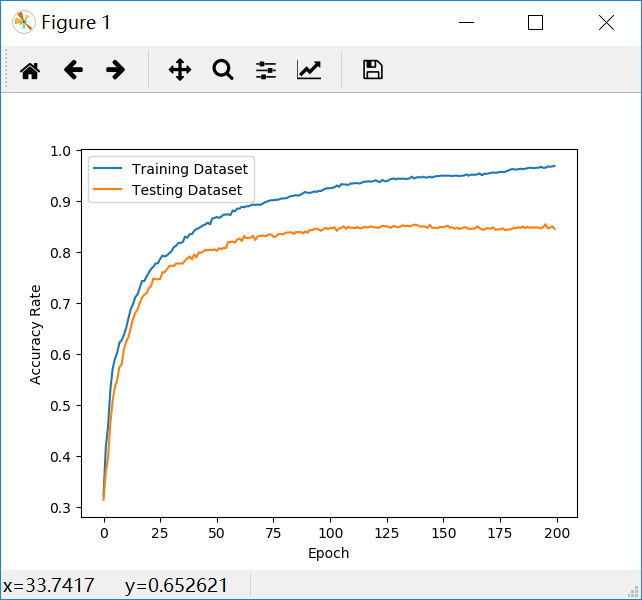
# MNIST对比

SSDBN中加入了proportion=0.5，其余参数一致

因为复现论文中未给出具体的参数，且运行一次时间过长，所以暂时未对参数进行优化，只是先大致对比下二者的差异，因此极有可能无法体现SSDBN优势

## SSDBN

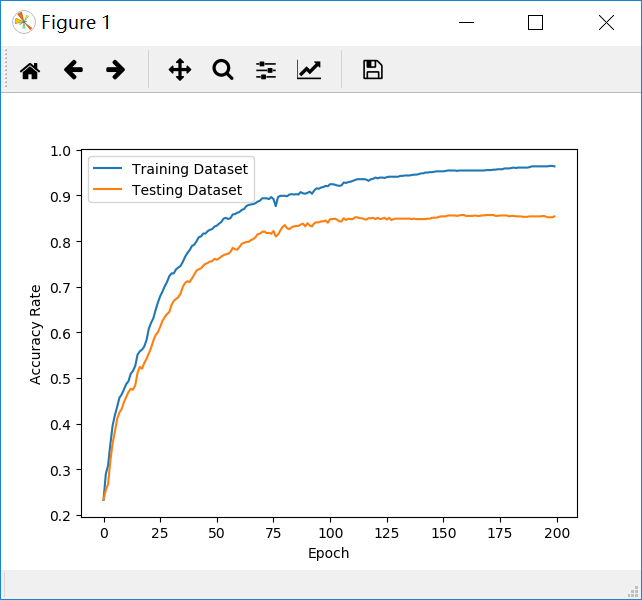
这里有一个问题：RBM的error在迭代中未出现明显下降，甚至上升，但是初始效果SSDBN比DBN好



训练集：0.9681818181818181

测试集：0.845

## DBN



训练集：0.9772727272727273

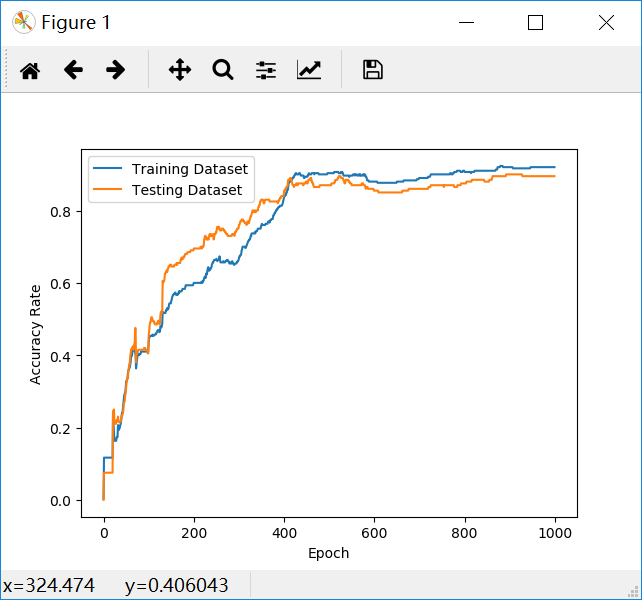
测试集：0.848

# 轴承数据集

SSDBN中加入了proportion，其余参数一致

存在问题，error的下降效果不明显，可能是参数不合适，考虑添加动量等进行优化

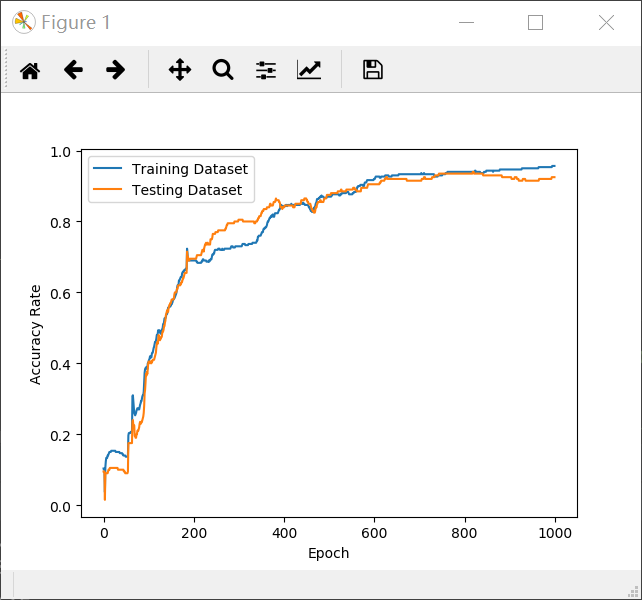
## SSDBN，proportion=0.05



训练集：0.92

测试集：0.895

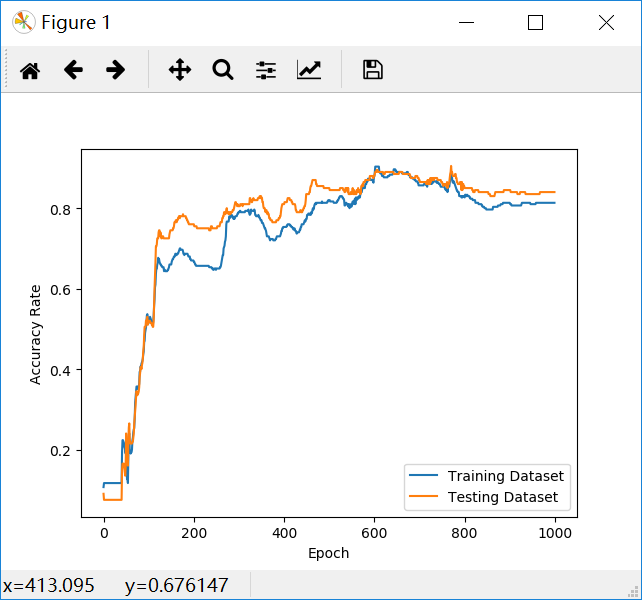
## SSDBN，proportion=0.1



训练集：0.9566666666666667

测试集：0.925

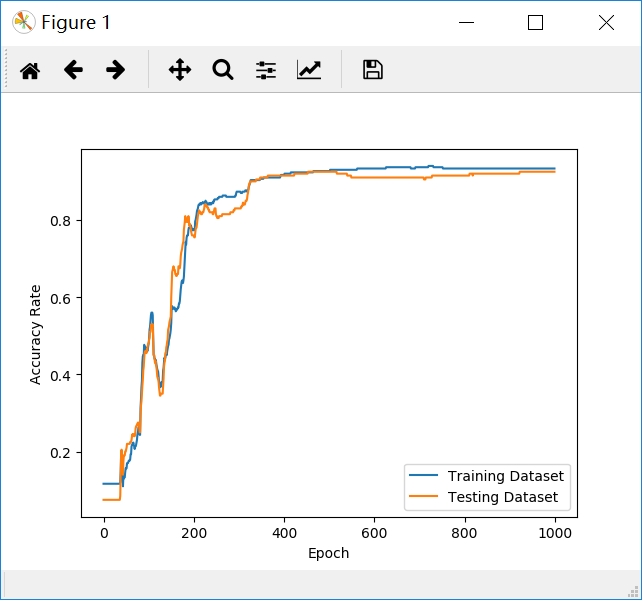
## SSDBN，proportion=0.3



训练集：0.8133333333333334

测试集：0.84

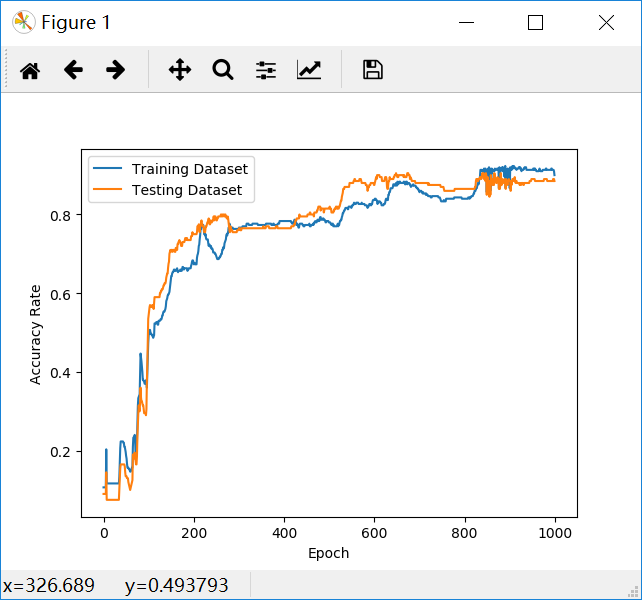
## SSDBN，proportion=0.5



训练集：0.9333333333333333

测试集：0.925

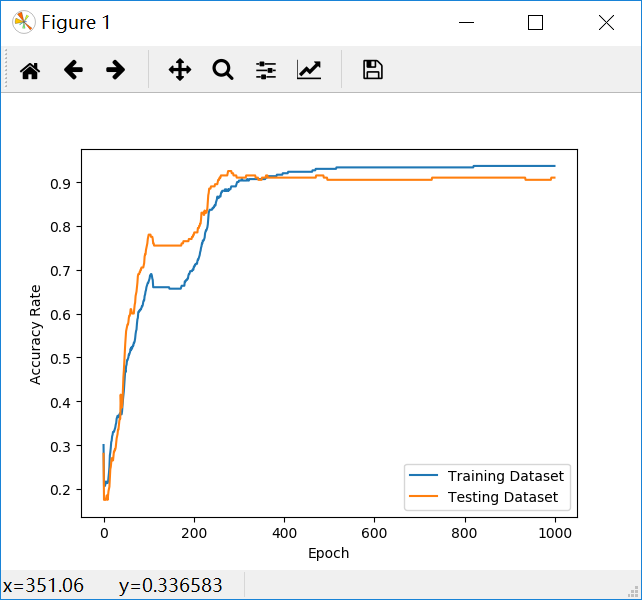
## SSDBN，proportion=1



训练集：0.9166666666666666

测试集：0.885

## DBN



训练集：0.9366666666666666

测试集：0.91

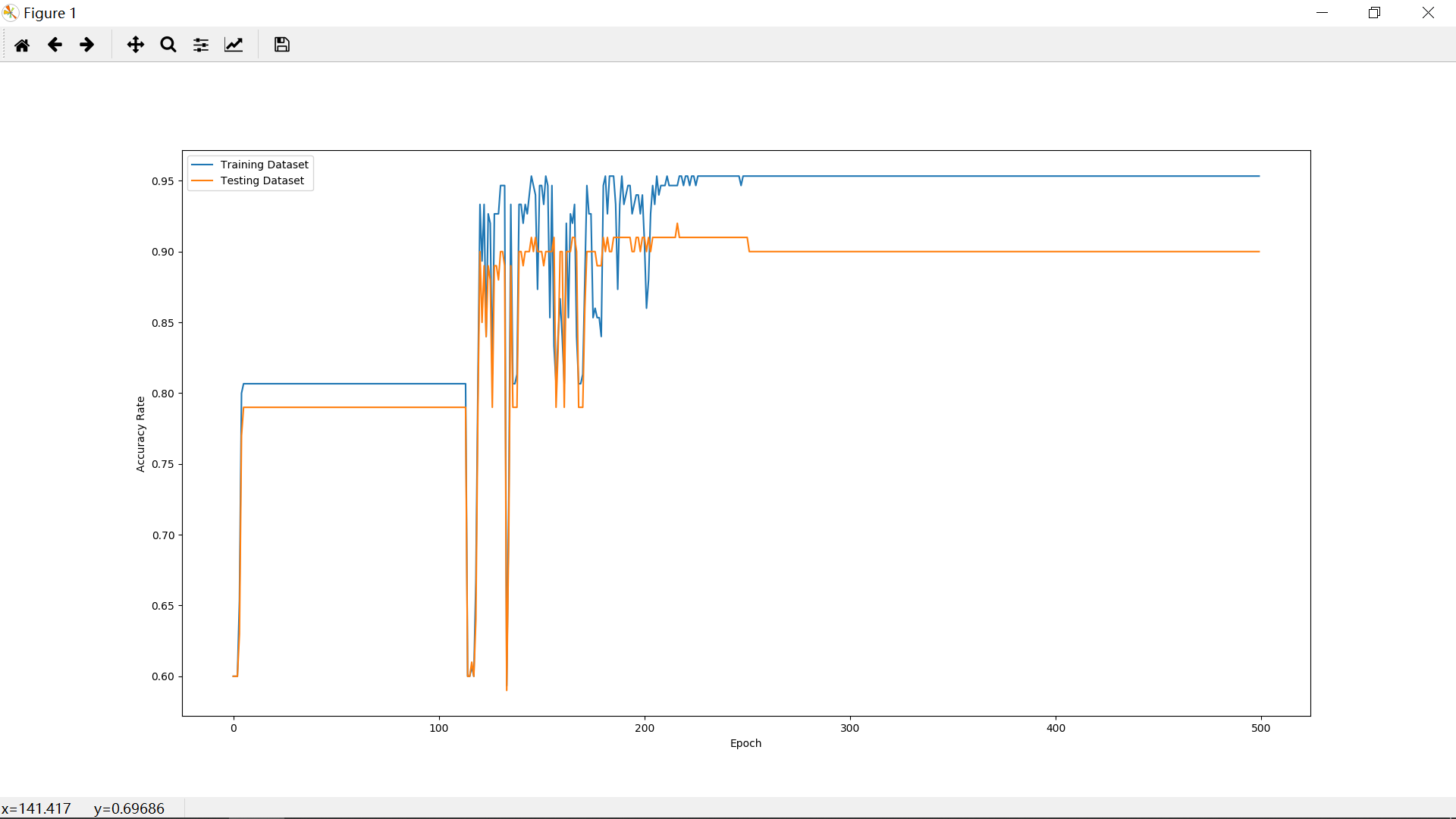
# 步骤1得到的数据集运行结果

因为步骤1运行时间较长，因此现在数据较少。

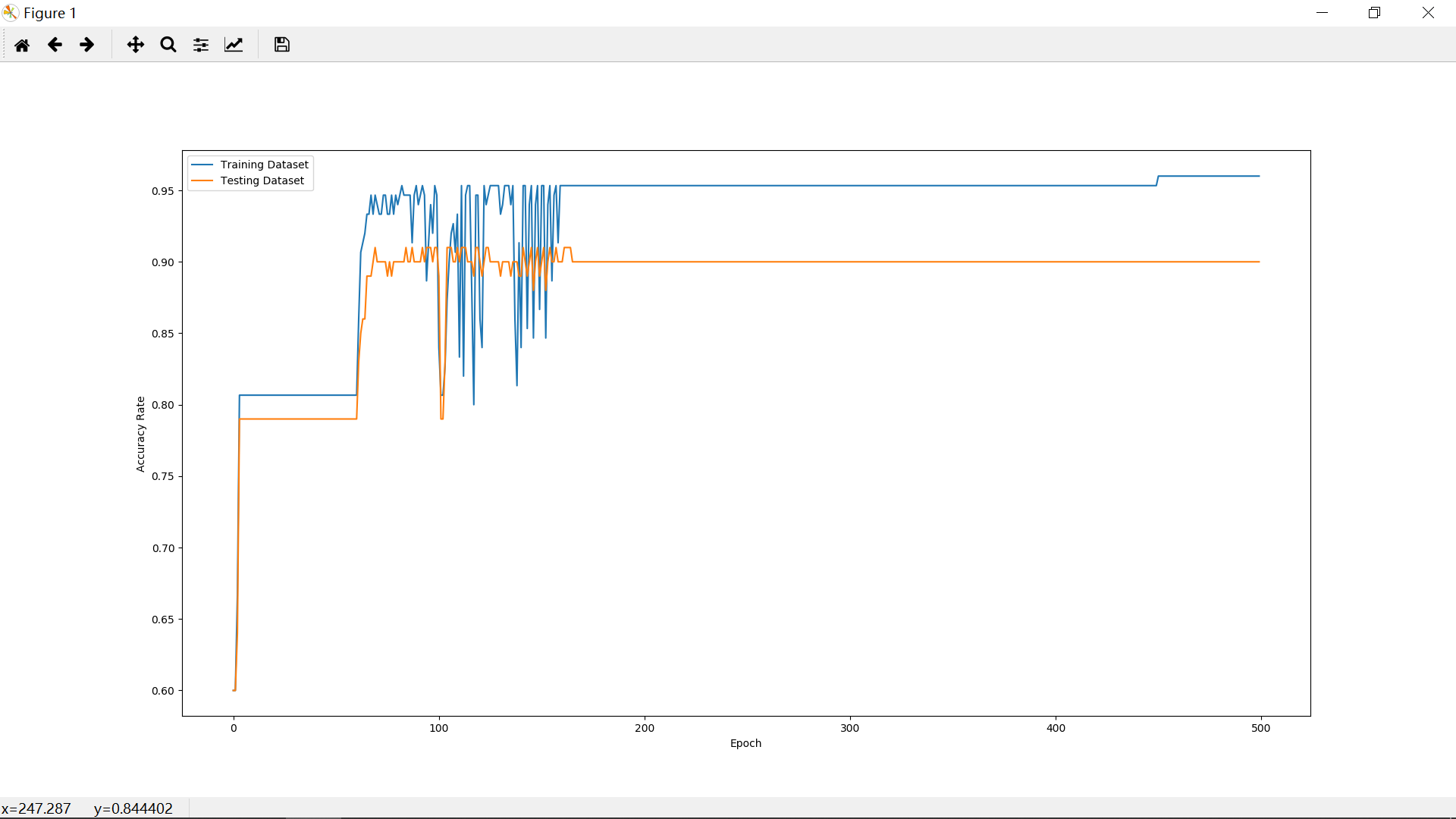
250条数据，40%的测试集，分为4类。

前半段动量为0.9，后半段动量为0.5。

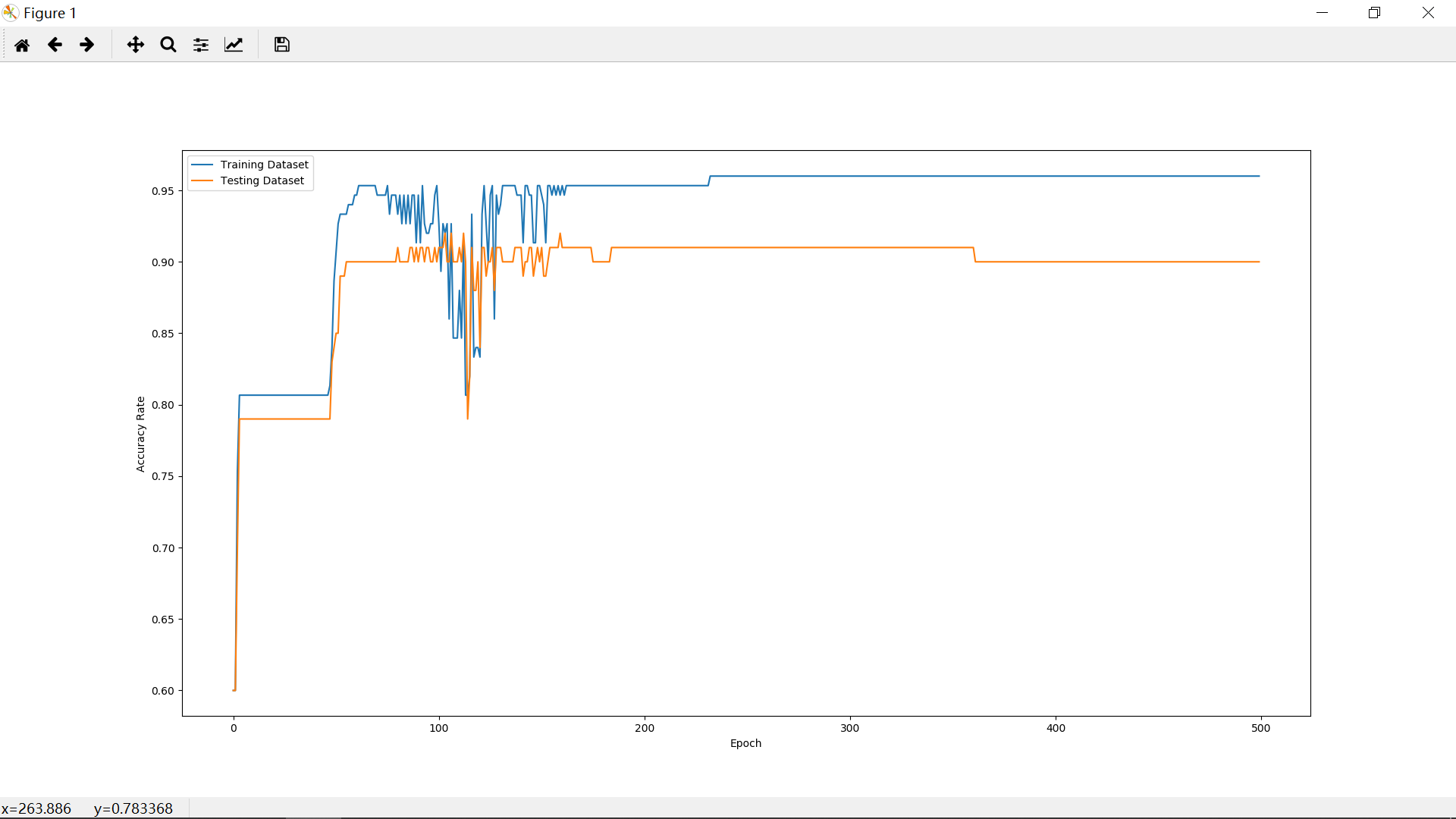
## Learning\_rate=1



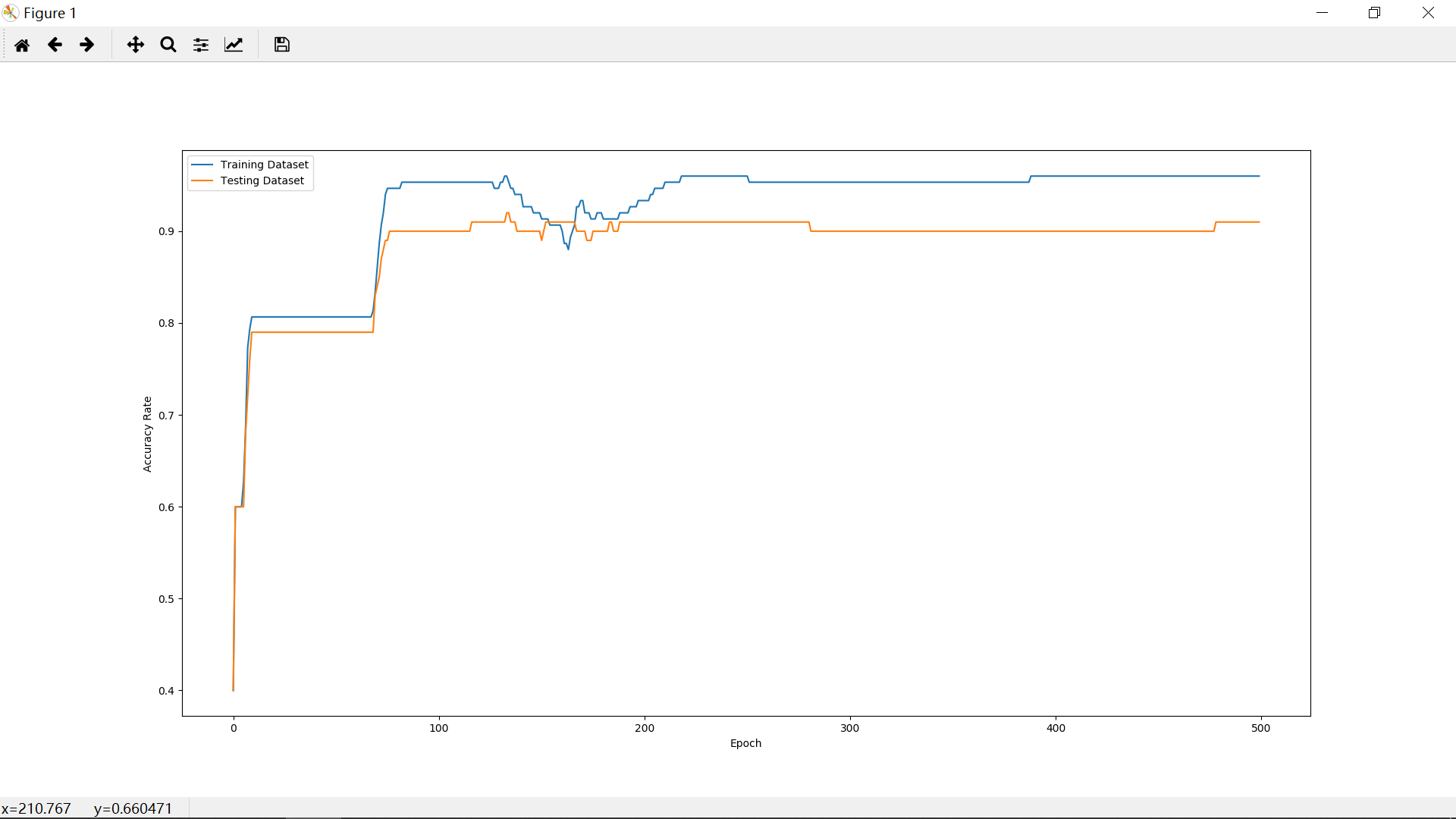
## Learning\_rate=0.5



## Learning\_rate=0.3



## Learning\_rate=0.1



最高正确率差异并不明显，训练集均在95.3%-96%之间，测试集在90%-91%之间。