

**《机器学习》课程实验报告**

**学 院 软件学院**

**专 业 软件工程**

**组 员**  **饶浩聪**

**学 号 201530612644**

**邮 箱 568302203@qq.com**

**指导教师**  **吴庆耀**

**提交日期** **2017年 12 月 8 日**

## 1. 实验题目: 线性回归和梯度下降

## 2. 实验时间：2017年 12 月 2 日

## 3. 报告人: 饶浩聪

## 4. 实验目的:

**进一步理解线性回归和梯度下降的原理。**

**在小规模数据集上实践。**

**体会优化和调参的过程。**

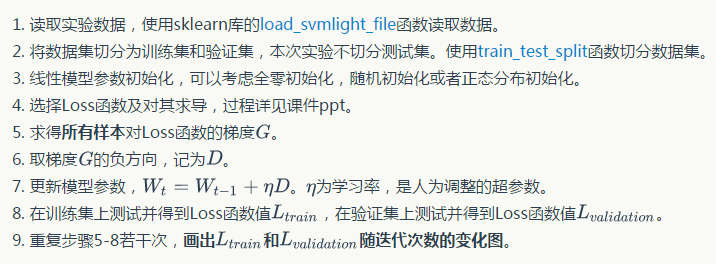
## 数据集以及数据分析：

**进一步理解线性回归和梯度下降的原理。**

**在小规模数据集上实践。**

**体会优化和调参的过程。**

## 6. 实验步骤:



## 7. 代码内容:

（针对线性回归和线性分类分别填写8-12内容）



本代码于2017.12.02(周六)当天完成，经过在本地IDE(Pycharm + py2.7)测试和jupyter-notebook(py3.6)测试无误，开发环境为python2.7，已将与python3.6不兼容的内容注释

## 选择的评估方法（留出法，交叉验证，k折交叉验证等）:

**线性回归和线性分类(SVM)都采用留出法进行验证**

**X\_train, X\_val, y\_train, y\_val = train\_test\_split(X, y, test\_size=0.33, random\_state=42)**

**验证集(留出)比例为训练集的1/3，并进行随机化**

## 模型参数的初始化方法:

**Lab1和Lab2均采用三种初始化方法：  
全零初始化，随机初始化或者正态分布初始化，但效果有所不同，会在之后分析**

**w = [0] \* 13**

**w = np.random.rand(13)**

**w = np.random.randn(13)**

**此处w0(w0\*x0, x0=1)采用偏置表示法，即另外用一个b来表示, b = w0**

## 选择的loss函数及其导数:

（以下内容可能与课程PPT有所不同，但为通用取法）

**线性回归选择的loss函数是：**



**其中，n为训练样本总数量，x(i)为第i个样本的x向量(总共有13个特征)**

**y(i)为第i个样本的预测结果**

**导数是：(累加后面的即为导数，此处给出权重w的更新公式)**

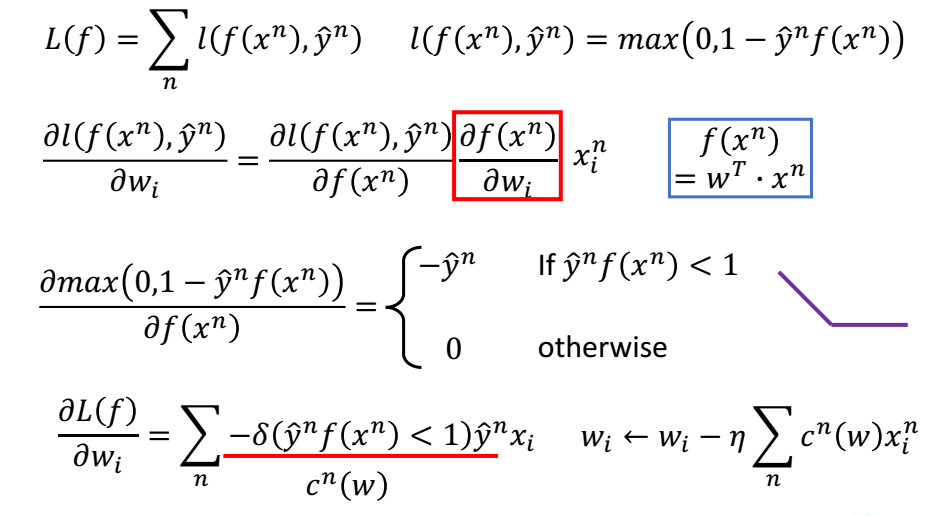
****

**其中，n为训练样本总数量，x(i)为第i个样本的x向量(总共有13个特征)**

**y(i)为第i个样本的预测结果，x(i)j为第i个样本的x向量第j个分向量，同理wj为w向量的第j个分向量，**

**线性分类选择的loss函数是：**

**下面是课程PPT里面关于SVM的loss函数(注意是总的，还没有取平均)及其求导,与我实际用的有所差别，可以对比一下**



**我实际所用的loss函数为：**



**实际上所用的w更新权重公式为(最右边为求的导)：**

****

**其中，n为训练样本总数量，x(i)为第i个样本的x向量(总共有14个特征)**

**y(i)为第i个样本的预测结果，x(i)j为第i个样本的x向量第j个分向量，同理wj为w向量的第j个分向量，f(x)为分类函数即wx+b,若大于0则y=1若小于0则y=-1**

参考资料：  
<https://zh.wikipedia.org/wiki/%E6%94%AF%E6%8C%81%E5%90%91%E9%87%8F%E6%9C%BA#.E6.AC.A1.E6.A2.AF.E5.BA.A6.E4.B8.8B.E9.99.8D>

<http://blog.csdn.net/liweibin1994/article/details/77511628>

<http://blog.csdn.net/v_july_v/article/details/7624837>

<http://blog.csdn.net/t0903/article/details/44014891>

## 实验结果和曲线图:

**Lab1线性回归：**

## 超参数选择（η,epoch等）：

## η取0.0016 训练迭代次数取200

## 评估结果（根据选择的评估方法）：

**三种参数初始化的方式最终loss都稳定在同一水平**

## 预测结果（最佳结果）：

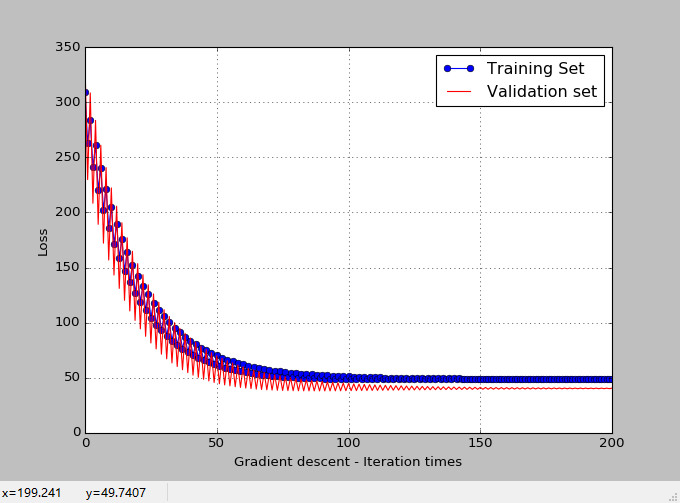
**在不同误差允许范围内可有不同的预测准确度，**

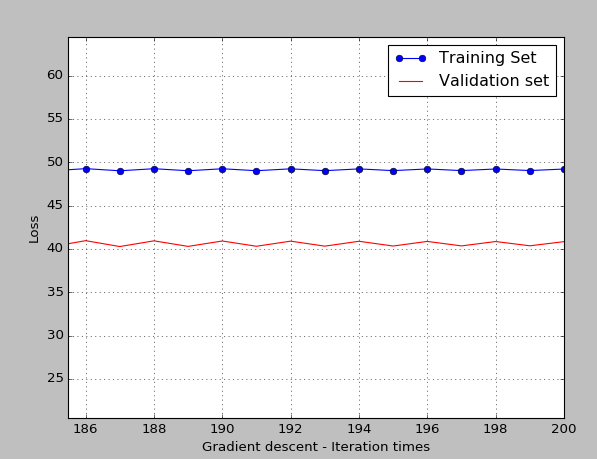
**若预测误差(|y实-y预|)<=10 准确率可以达80%以上**

**若预测误差(|y实-y预)<=20 准确率可以达90%以上**

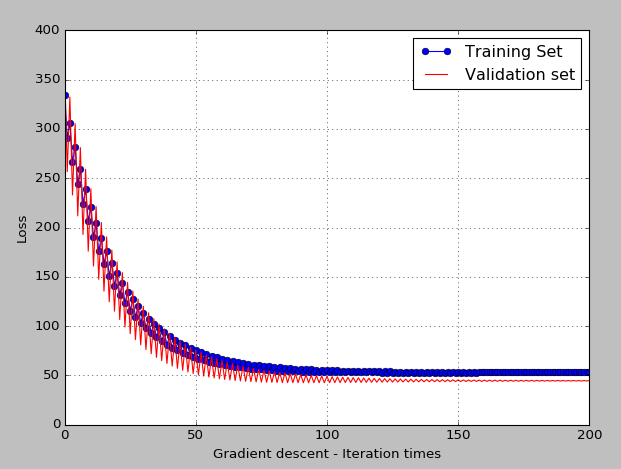
## loss曲线图：

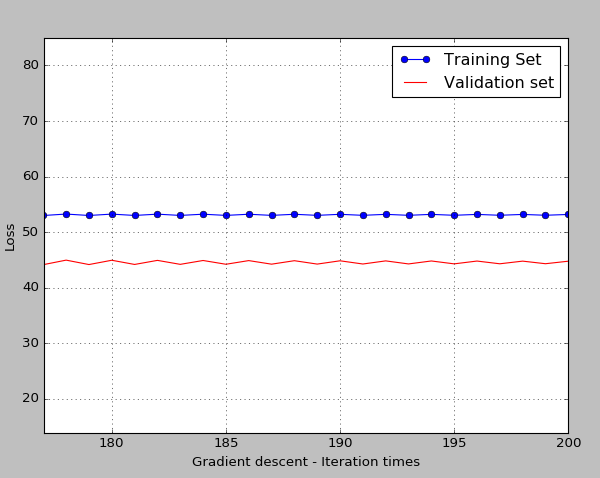
1. **模型w参数全0初始化**



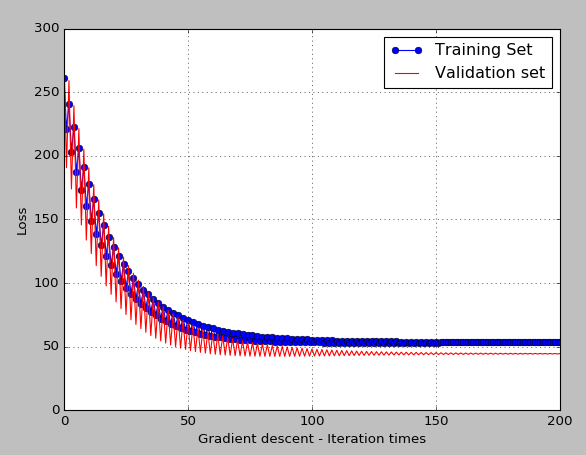


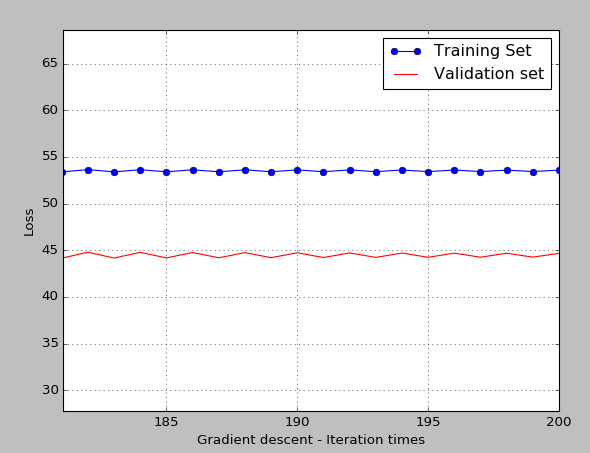
**B.模型w参数随机初始化**





**C模型w参数高斯分布初始化**





**Lab2线性分类：**

## 超参数选择（η,epoch等）：

## η取0.0096 训练迭代次数取40-300之间，不以训练次数为标准停止训练，而是以loss函数衰减进行到一定程度时停止，此处取0.00001(变化程度小于0.00001则停止训练)

**(实际代码中已经根据训练结果自动确定好训练次数)**

## 评估结果（根据选择的评估方法）：

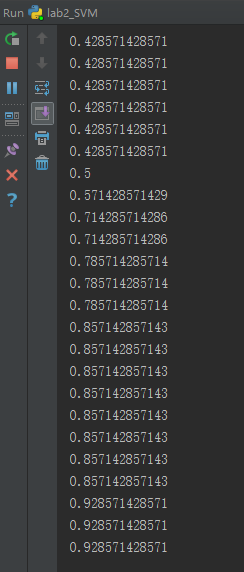
**三种参数初始化的方式最终loss有比较大的差异**

**随着loss减少，该模型在全部样本上的分类准确率也随之上升**

## 预测结果（最佳结果）：(这里取截图

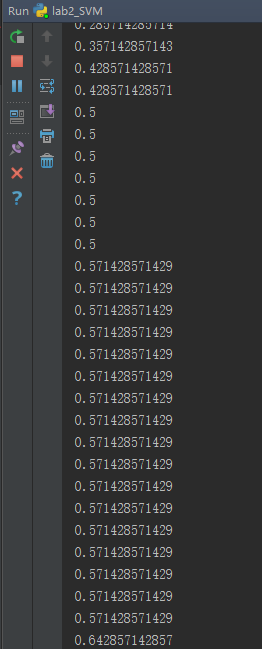
1. **模型参数w全0初始化，学习系数为0.096，训练次数为50的时候**

**在所有样本预测的准确率的变化情况，可以看到最后准确率已经达到92.8%**



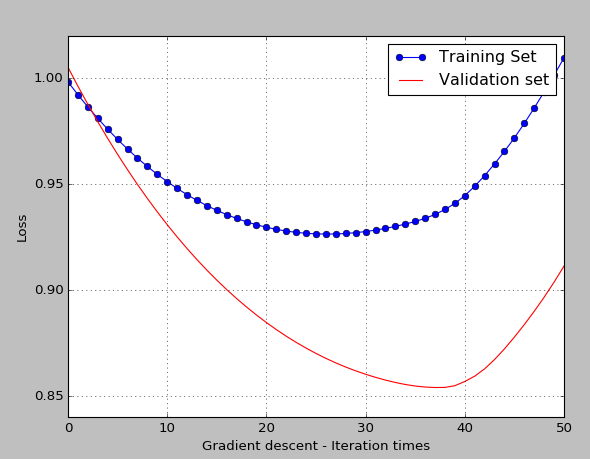
1. **模型参数w高斯分布初始化，学习系数为0.096，训练次数为200的时候**

**在所有样本预测的准确率的变化情况，准确率达到64.2%，但是随之loss的减少，在250-300的训练过程中，预测准确率会下降，因为出现了过拟合的问题，可见，在训练过程中模型的效果要取得最好，就要防止过拟合，训练次数控制在一定的范围内，否则就算loss减少到很低，在测试集上的预测效果也不佳。**

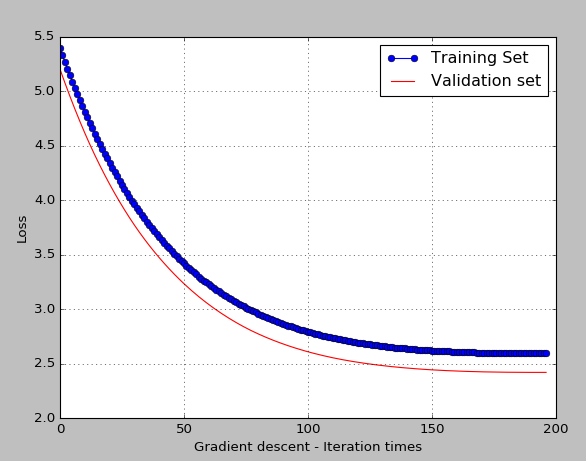


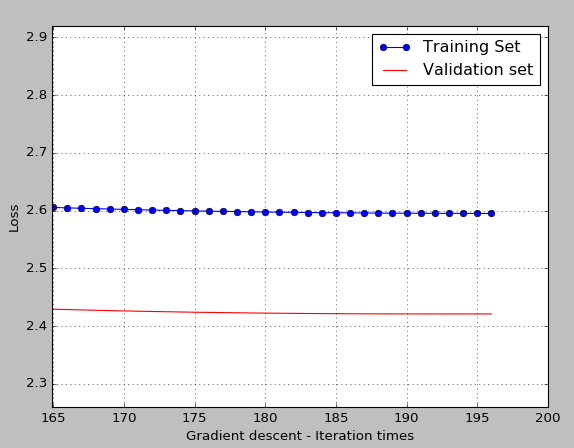
## loss曲线图：

**A.模型w参数全0初始化**

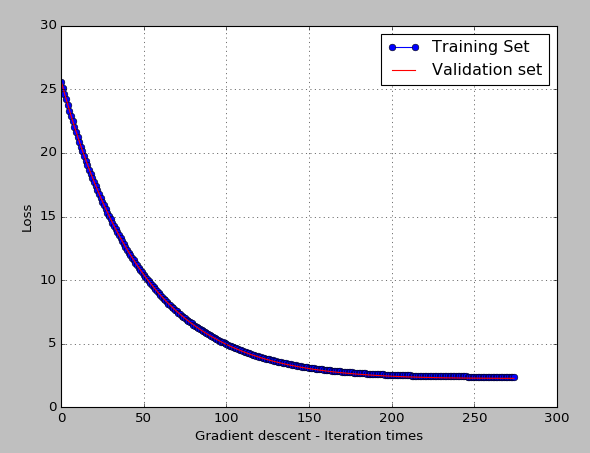


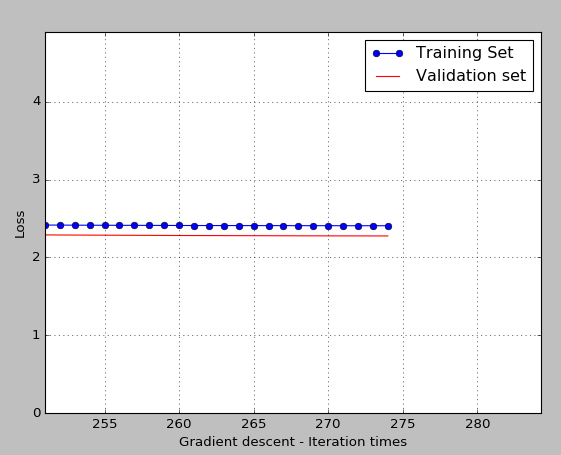
1. **模型参数w随机初始化**





1. **模型参数w高斯初始化**





## 实验结果分析:

**Lab1线性回归：**

**实验结果表明，**η学习率不能取过大，否则容易导致无法收敛，在本次实验中，η分别取了0.1/0.2/0.16/0.016进行测试(保证模型其它参数一致)，最终发现当η取0.016的时候收敛效果最佳；本次实验模型参数w用了三种方式进行初始化，在最佳η情况下，三种方式所得到的loss衰减曲线是基本持平的，所以在此次线性回归模型中w初始化的方式可以有多种，效果差不多。

**Lab2线性分类：**

**实验结果表明，在**η取合适的情况下(即在所有参数初始化方式下都能取得loss正常衰减速度的情况下)，(本次实验分别取了0.1/0.96/0.096/0.2，最终确定为0.096），SVM线性分类效果与模型参数初始化有一定联系，在实验过程中，当w全0初始化时，预测准确率最高，而且不需要很多的训练次数；在高斯初始化和随机初始化的情况下，在验证集上的预测准确率明显比较低，而且虽然loss在减少，但是会出现过拟合的现象，使得在训练集上效果好，在验证集上效果差，所以在实际应用中一定要选择合适的训练次数和loss阈值来避免过拟合

## 对比线性回归和线性分类的异同点：

1.Linear SVM和LR都是线性分类器

2.Linear SVM不直接依赖数据分布，分类平面不受一类点影响；LR则受所有数据点的影响，如果数据不同类别strongly unbalance一般需要先对数据做balancing。

3.Linear SVM依赖数据表达的距离测度，所以需要对数据先做normalization；LR不受其影响

4.Linear SVM依赖penalty的系数，实验中需要做validation

5.Linear SVM和LR的performance都会收到outlier的影响

## 14.实验总结：

关键一点：模型理解很重要，只有先理解无误，才能够灵活运用

更关键一点：模型使用中要多实践，调参不是一蹴而就的事情，要多尝试，最好通过调试输出相关数据(例如loss和预测准确率)来判断模型是否已经训练出效果，如果相关优化方法没能取得比较好的效果，要注意是否正确地输入模型的公式，是否有出现过拟合的现象们是否没有遵循这个模型本来的限制。

实验记录和实验总结是比实验更重要的事情，有助于加深理解，而且有了记录之后下次如果忘了还可以查阅。

**实验过程中参考的资料：**  
<https://zh.wikipedia.org/wiki/%E6%94%AF%E6%8C%81%E5%90%91%E9%87%8F%E6%9C%BA#.E6.AC.A1.E6.A2.AF.E5.BA.A6.E4.B8.8B.E9.99.8D>

<http://blog.csdn.net/liweibin1994/article/details/77511628>

<http://blog.csdn.net/v_july_v/article/details/7624837>

<http://blog.csdn.net/t0903/article/details/44014891>

<http://www.jianshu.com/p/738f6092ef53>

<https://www.zhihu.com/question/26768865>

<http://blog.csdn.net/t0903/article/details/44014891>