

**《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 15 日**

## 1. 实验题目: 逻辑回归、线性分类与随机梯度下降

## 2. 实验时间：2017年 12 月 10 日

## 3. 报告人: 饶浩聪

## 4. 实验目的:

对比理解梯度下降和随机梯度下降的区别与联系。

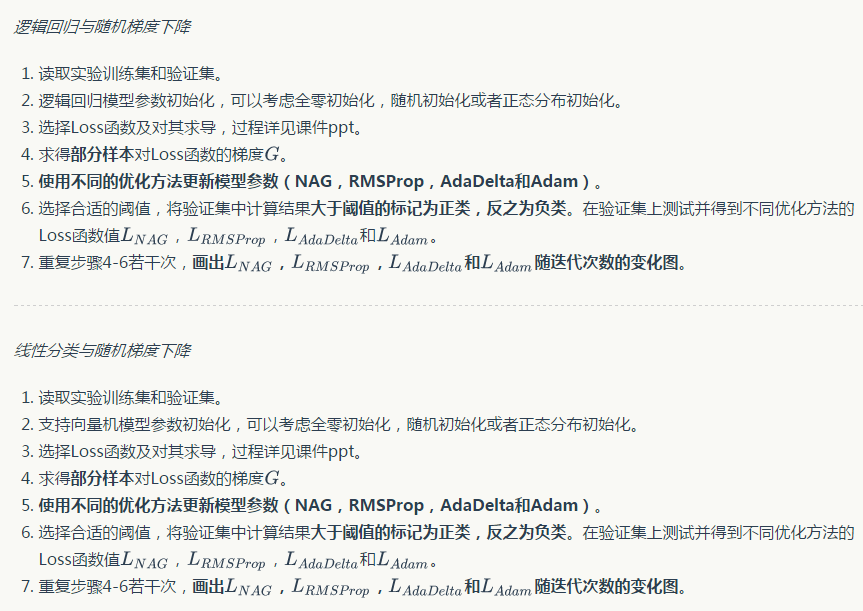
对比理解逻辑回归和线性分类的区别与联系。

进一步理解SVM的原理并在较大数据上实践。

## 数据集以及数据分析：

实验使用的是LIBSVM Data的中的a9a数据，包含32561 / 16281(testing)个样本，每个样本有123/123 (testing)个属性。请自行下载训练集和验证集。

## 实验步骤:



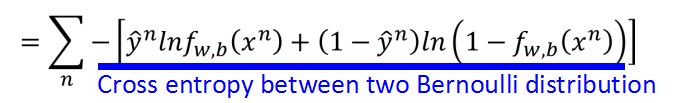
## 7. 代码内容:

（针对逻辑回归和线性分类分别填写8-11内容）

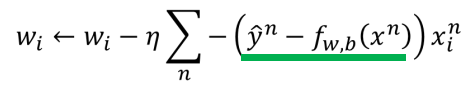
## 8. 模型参数的初始化方法:

9.选择的loss函数及导数：

**1.逻辑回归选择的loss函数是(对数损失函数)：**

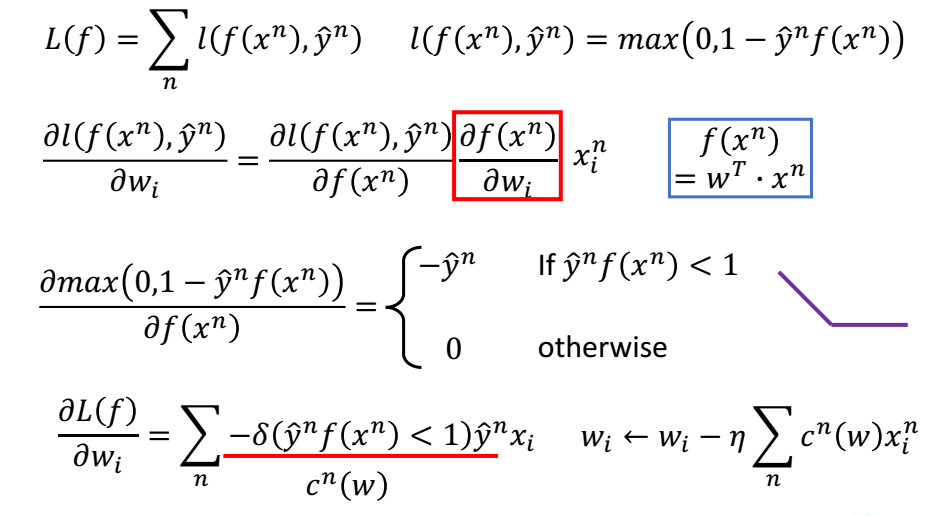


**权重更新的公式是(包括导数)：**



**2.线性分类SVM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**

## 实验结果和曲线图:（各种梯度下降方式分别填写此项）

## 超参数选择：

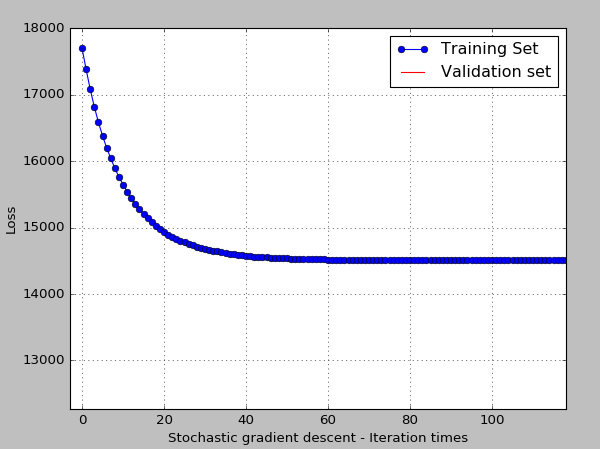
逻辑回归和线性分类实验中，w采用全零初始化(经过测试发现这个效果最佳)，然后全局学习参数(若有)，逻辑回归为a = 0.0010，SVM为aa = 0.00096

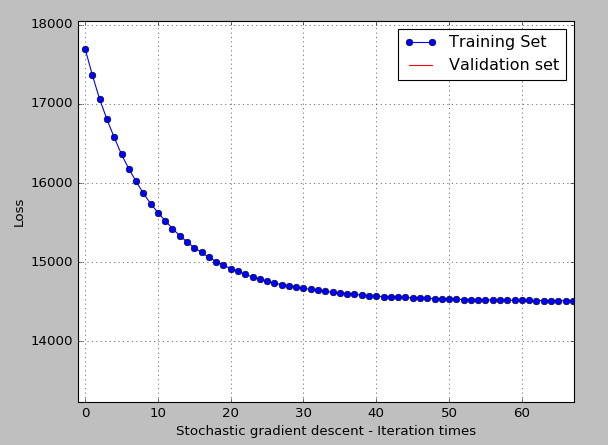
## loss曲线图：

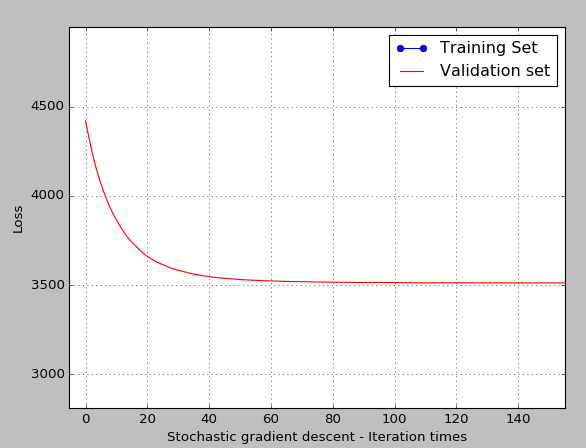
分别如下所示(调试时使用Pycharm输出，与jupyter输出的图像风格不太一致，在此说明一下)

**SGD**

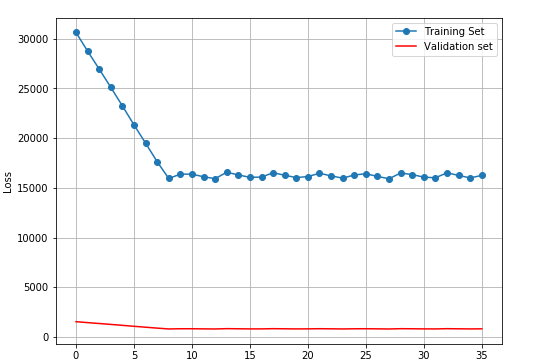
**逻辑回归：**



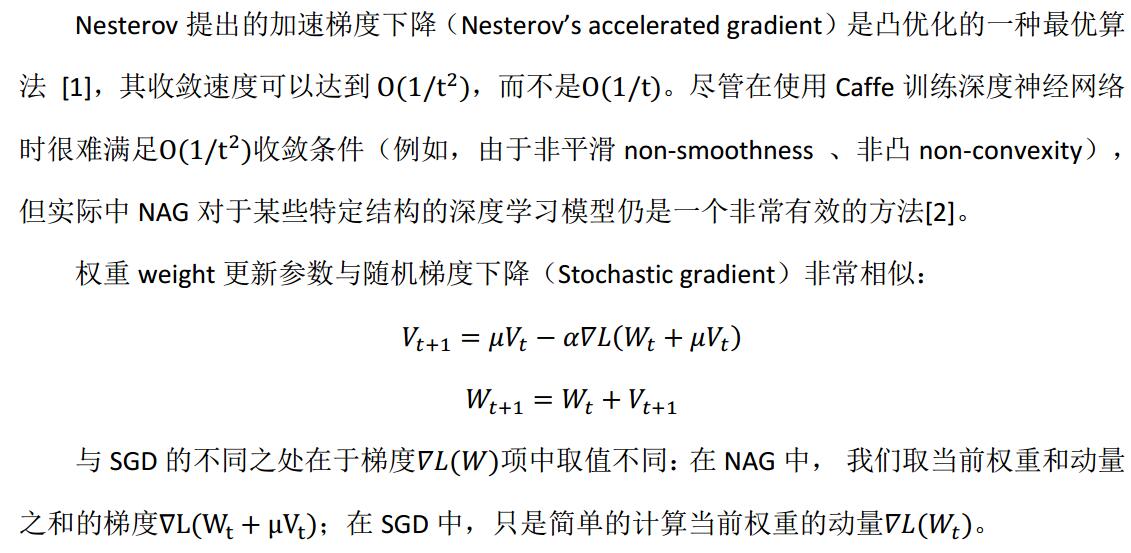




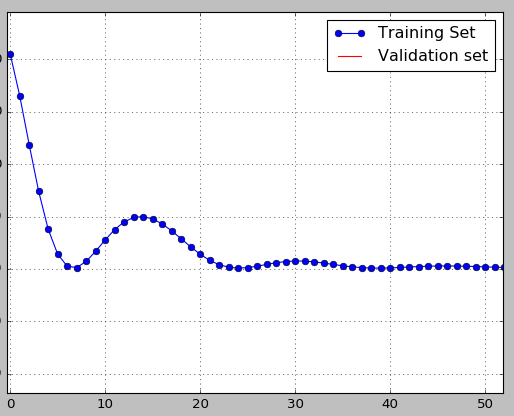
**SVM：**



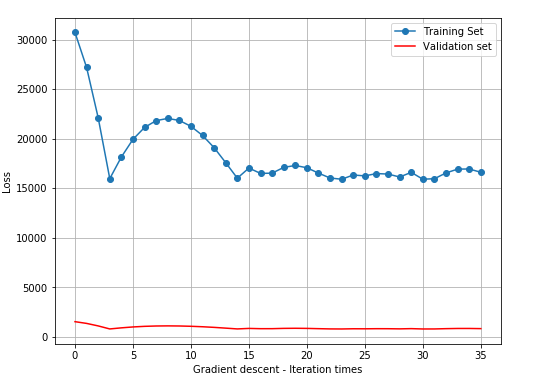
#### ****Nesterov accelerated gradient****



**逻辑回归：**

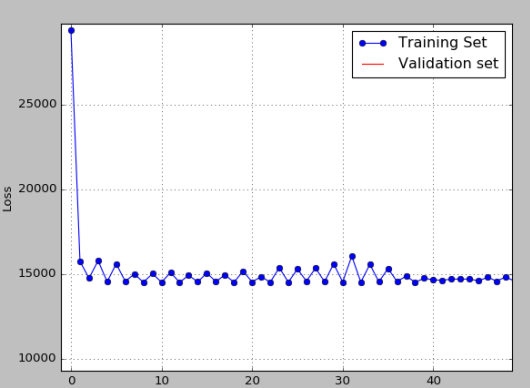


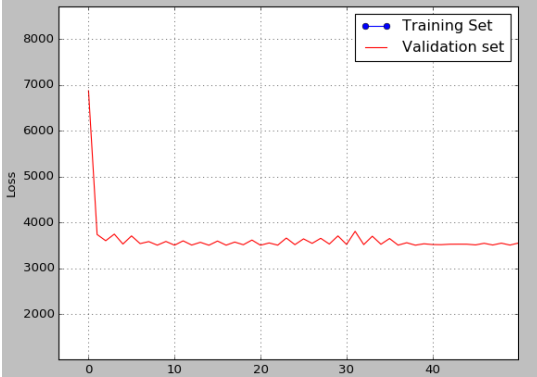
**SVM：**



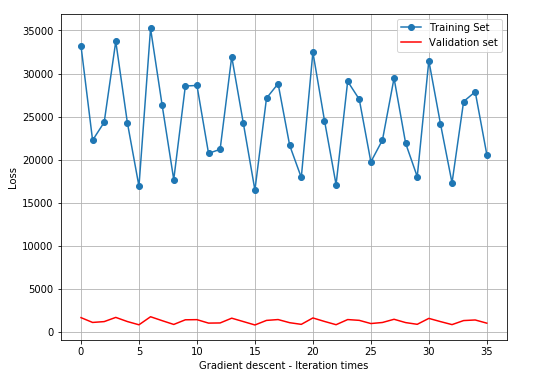
**RMSprop**

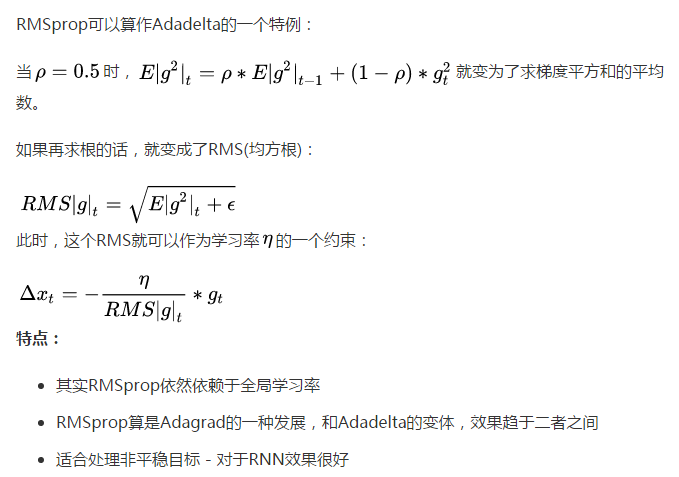
**逻辑回归：**





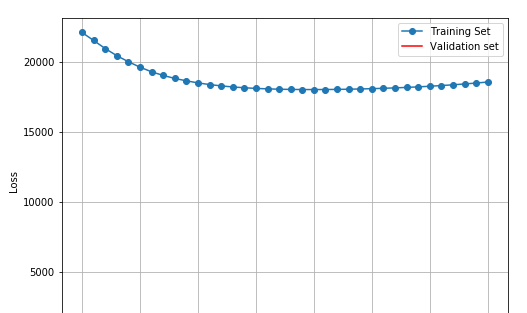
**SVM：**



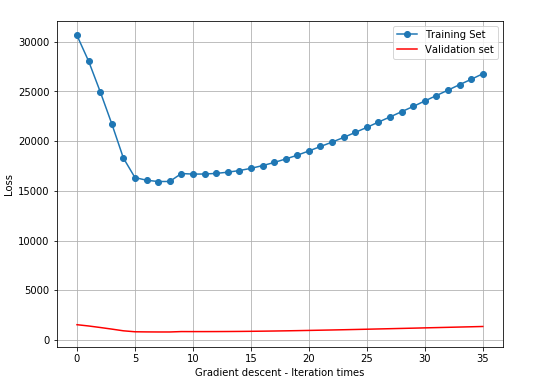


1. **Adam**

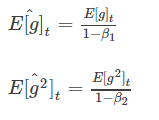
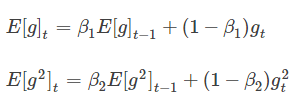
**逻辑回归：**

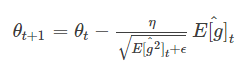


**SVM：**



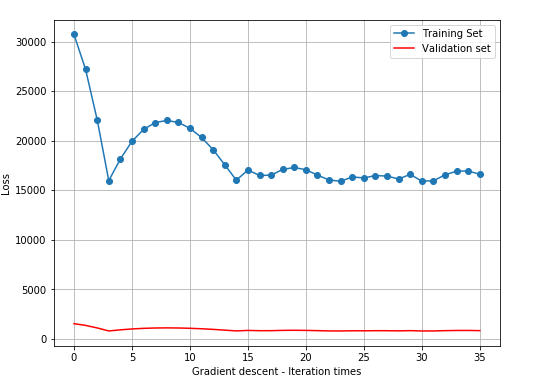
Adaptive Moment Estimation (自适应矩估计Adam)是另外一种为每个参数提供自适应学习率的方法。   
同RMSprop、Adadelta相比，Adam在对梯度平方(二阶矩)估计的基础上增加了对梯度(一阶矩)的估计，使得整个学习过程更加稳定。





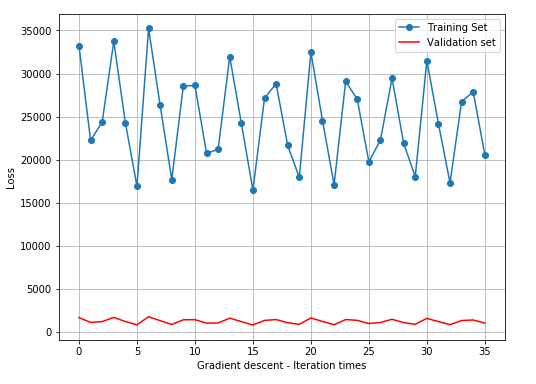
通常 β1,β2分别被设置为0.9和0.999.

## 实验结果分析:

NAG 

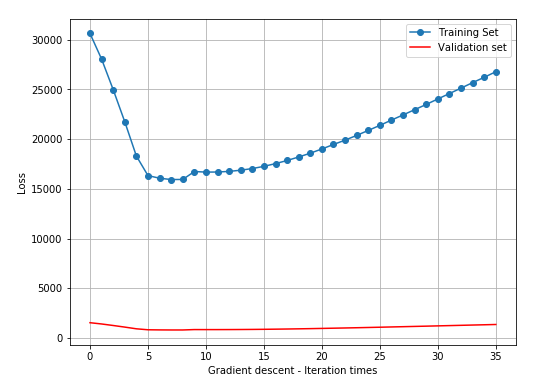
NAG学习参数的更新可以自动调整学习速度，如图所示，刚开始速度加快(还没收敛)，到后来的震荡程度越来越小，在实际应用中能够增大跳出局部最优值的可能性。

**RMSProp**



RMSprop 无论是否达到收敛点，都会不断地震荡，即不断更新方向寻找更优的收敛点，越到后面可以明显看出越接近全局最优点。

**Adam**



Adam在前期大幅加快了随机梯度下降的速度，后期则放慢速度，若陷入局部最优点之后就比较难以出来，从图像可见，已经往错误的梯度方向继续进行。

## 12.对比逻辑回归和线性分类的异同点：

**相同点：**

第一，LR和SVM都是分类算法。

1. 如果不考虑核函数，LR和SVM都是线性分类算法，也就是说他们的分类决策面都是线性的。LR也是可以用核函数的。原始的LR和SVM都是线性分类器

第三，LR和SVM都是监督学习算法。

第四，LR和SVM都是判别模型。

判别模型会生成一个表示P(Y|X)的判别函数（或预测模型），而生成模型先计算联合概率p(Y,X)然后通过贝叶斯公式转化为条件概率。简单来说，在计算判别模型时，不会计算联合概率，而在计算生成模型时，必须先计算联合概率。或者这样理解：生成算法尝试去找到底这个数据是怎么生成的（产生的），然后再对一个信号进行分类。基于你的生成假设，那么那个类别最有可能产生这个信号，这个信号就属于那个类别。判别模型不关心数据是怎么生成的，它只关心信号之间的差别，然后用差别来简单对给定的一个信号进行分类。常见的判别模型有：KNN、SVM、LR，常见的生成模型有：朴素贝叶斯，隐马尔可夫模型。当然，这也是为什么很少有人问你朴素贝叶斯和LR以及朴素贝叶斯和SVM有什么区别（哈哈，废话是不是太多）。

第五，LR和SVM在学术界和工业界都广为人知并且应用广泛。

**LR和SVM的不同点：**

第一，本质上是其loss function不同。

第二，支持向量机只考虑局部的边界线附近的点，而逻辑回归考虑全局（远离的点对边界线的确定也起作用）。

支持向量机改变非支持向量样本并不会引起决策面的变化

逻辑回归中改变任何样本都会引起决策面的变化​

线性SVM不直接依赖于数据分布，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。​

第三，在解决非线性问题时，支持向量机采用核函数的机制，而LR通常不采用核函数的方法。​

第四，​线性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

1. SVM的损失函数就自带正则(损失函数中的1/2||w||^2项），

## 13.实验总结：

**1.通过本次实验，既掌握了随机梯度下降的方法（在实际应用中采用随机mini-batch的方法进行梯度的更新项计算，在本次实验中采取样本数量的1/8进行随机更新，详情可见代码），又尝试了很多不同的学习参数更新方法，比如RMSprop, Adam, NAG,每一种方法都有优劣，在实践的过程中，通过对Loss图像的分析，进一步领悟了各种学习参数更新方法的特性，有助于以后机器学习的深入开展。**

**2.本次实验在实验一的基础上对代码进行了效率上的提高，实验一采用是计算方式是python自带的list，效率不高，实验二和实验三开始全部使用numpy的计算，例如矩阵的计算numpy.dot(x,y)，还有log函数的计算numpy.log等，在一定程度上减少了循环的次数，可以达到一次高效计算实现目的的效果。**

**3.进一步总结了LR和SVM的异同，更深入地思考了关于不同Loss函数之间的对比(例如逻辑回归的方差loss函数和对数loss函数的优劣)，对于以后loss函数的选择具有重要的意义，同时加强了对大规模数据处理的分类能力。**