



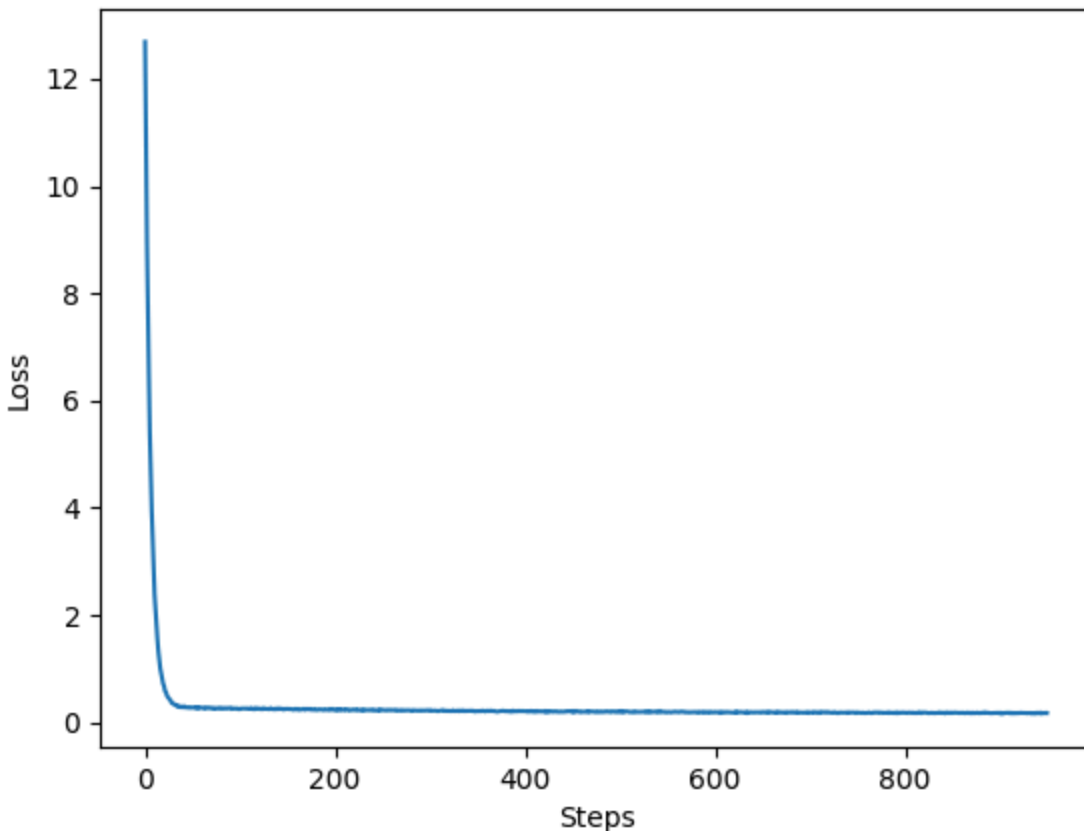
lab1_report

龙涵睿PB22111606

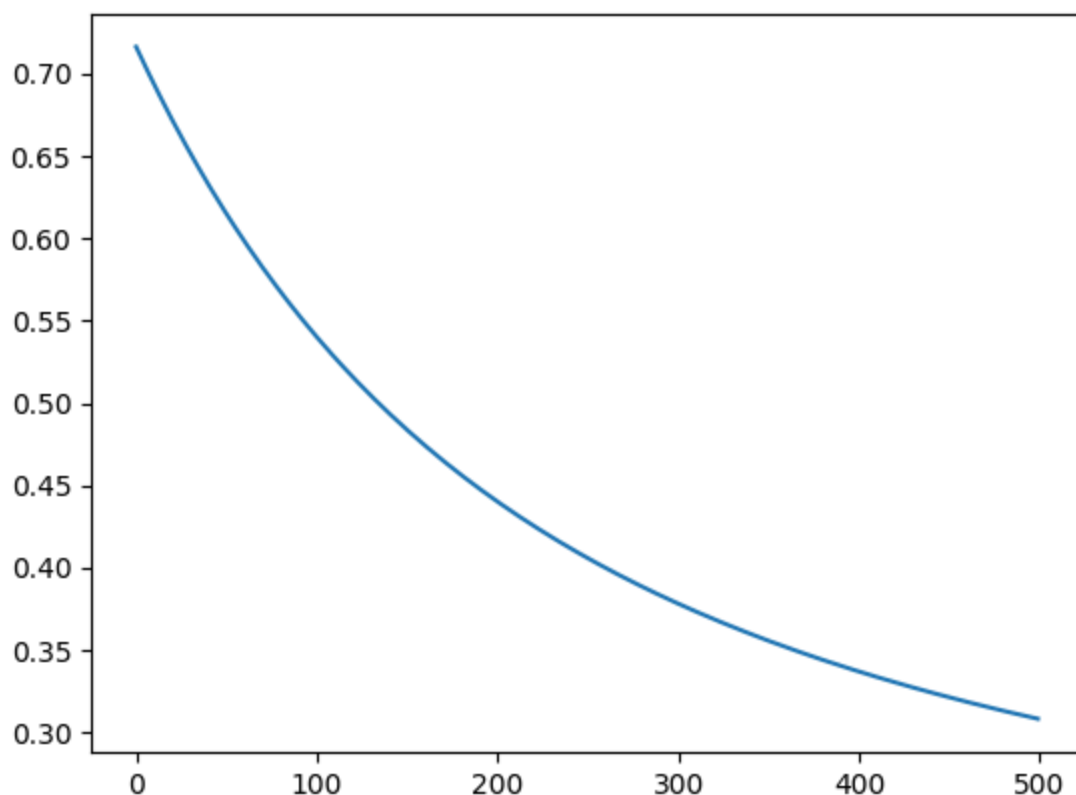
1.实验流程

先看了一遍文档，然后开始看实验，难受的是一开始在仓库里面没找到要用哪个数据集，导致我还以为都可以用，然后到处看用哪个好，后面才看到有确定的数据集
在写实验过程中大量查阅官方文档，一开始还没想到要对数化Run_time，后面在.ipynb上面分析了一下分布才知道

2.loss曲线和调参过程



这个是线性回归的loss曲线，我一开始的init值就选得还不错，没有差特别多，然后把lr狠狠提了一下，看着曲线感觉还可以，error情况也还可以，加上国庆节，就没接着往后狠狠卷。



这个是二元分类的loss曲线

之前相同的steps结果acc能差0.3几，后来发现是init的时候差的有点多，在凸优化问题中真的选择大于努力，lr不够大的情况下都跑完了可能还没别人的init的weight的得分高。。。

3.最好的结果

感觉和交上来的差不多啊，训练新的然后之前的png就没了，有个图还挺好看的，在跑二元分类的最好结果就是 $acc=1$ 。

看issue明白了，之前我在对数化的时候多写了一个小偏置，导致label分布更不均匀了，影响了后面的训练，eval也说acc是1，我受宠若惊

4.1 预处理的额外操作

我把Run_time对数化了，因为我觉得对数化过后数据分布更加平均了，误差也会更小

4.2 交叉熵损失的优势

交叉熵损失在二分类问题上收敛快，对错误的结果惩罚力度大，对我错误的init的nan力度也大

4.3 调参经验

调参前先热身一下，活动手指，在等待training的时候狠狠铲一把，顺便看看loss表现怎么样，拿大lr和小lr夹逼一下，我在utils里面把梯度那一块加上了几条print函数给我实时汇报当前的grads情况，如果正负横跳那就是lr大了，如果loss降得过慢那就是lr小了
epochs和batchsize我没怎么调，batchsize我用了几个感觉差不多就用默认的了，epochs就看loss.png里面后面都差不多平了就不用烤鸡硬跑了

4.4 正则化

报告，我没做，因为我觉得不是那么必要吧，不用正则化也能跑出来的说不定我把正则化一加上程序和我都跑不了了

5. 花费时间

Project Summary (Last updated on Mon, Oct 14th 9:34pm)

USTC-ML24-Fall: Total
编辑器使用时间: 5.4 小时
活跃编程时间: 1.7 小时
添加的代码行数: 312
删除的代码行数: 74
总的键入数: 6,062

好了，这下都知道哥们瞪半天写不出来了
实际上花费时间还要长一点，插件下的晚，加之前的大概十五个小时吧

6. More

本来是在国庆节就写完了的，但在周一发现了超级大bug，感觉不重新提交怕是连扣十分都不如，故又交一次