

Your first neural network

此部分属于 Deep Learning Nanodegree Foundation Program

与大家分享你取得的成绩！

Requires Changes

还需满足 2 个要求 变化

模型实现的很不错，超参数调整还有一些问题，我给你了一些建议～你可以尝试看看～加油！

代码功能性



notebook 上的所有代码都可以在 Python 3 下成功运行，并且通过所有的单元测试。

很好，你的代码通过了所有的单元测试！



sigmoid 激活函数执行正确

非常好，你使用了lambda函数，你也可以定义一个函数，不过稍显麻烦，两个方法都可以。但是如果使用如下方法而不是lambda函数，就有一个好处，当你想要更改激活函数的时候，可以很方便，而且如果激活函数比较复杂，这样的定义方法反而更加简洁。

```
def sigmoid(x):  
    return 1/(1+np.exp(-x))
```

正向传播



隐藏层的输入在训练和运行函数中均正确执行

非常好，这一步你实现的很对。



隐藏层的输出在训练和运行函数中均正确执行



输出层的输入在训练和运行函数中均正确执行



网络的输出在训练和运行函数中均正确执行

很好，你能正确的判断是否需要加上激活函数，记得如果是分类问题的话，激活函数还是需要的。因为如果我们要做分类的话，激活函数可以将得到的数值“压缩”到0-1之间，这样我们就可以当作是计算概率啦。但是我们这个例子中，并不是做分类，而是预测一个连续的数值，所以没有必要加上激活函数哦。

反向传播



网络的输出误差执行正确。



对两个权重的更新执行正确。

很好，反向传播对于刚接触的人来说有点容易弄糊涂，因为权重太多，而且以矩阵的形式运算，很容易弄混淆。你这里实现的很不错。

超参数



对迭代次数的选择满足：能让训练网络作出准确的预测，且不会对训练数据产生过度拟合。

2000的迭代次数有点低哦，我们建议在5000-8000左右，这样能够观察loss曲线的形态变化，是否保持水平，是否降到了足够低。



对隐藏层单元个数的选择满足：能让网络准确预测单车人数，使其具有泛化能力，但不会过拟合。

隐藏层单元的个数是一个非常开放的问题，每个人都有自己的参数取值。有时候这个参数的取值有点“玄学”的感觉。但是实际上还有一些基本的规则的，比如单元的个数应该介于input 单元个数（52）和output单元个数（1）之间。我比较推荐至少 8-10 个才能很好的拟合我们的数据。我个人尝试的结果来看，10-14个左右能够获得比较好的模型性能。这里有篇 [文章](#) 介绍了一些选择隐藏层单元个数的经验，供你参考。



对学习速率的选择满足：能使网络成功收敛，且仍然具有较高的时间效率。

学习速率实际上用来更新权重，而我们的例子中，更新权重的时候除以了n_records，所以0.1的learning rate有点小了，我个人推荐0.5-0.8都不错。可以更快的收敛模型，也不会耗费太久的时间训练模型。