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！

Meets Specifications

很好，你的validation loss算是学生中比较低的了，你的超参数调整的也很不错。总的来说完成的很好！一些调整参数的技巧我在前面已经和你说过了，主要是根据loss曲线进行判断。也有人喜欢用grid search的方法来取超参数，但是这也要建立在你对超参数的大概取值范围有一定的熟悉程度的情况下才能够使用这种方法，不然你的超参数搜索空间会很大。

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。

这是测试驱动的开发的一个好例子。一般来说你会在写实际代码前先完成测试代码，确保你写的代码能够完成你想要它完成的工作。这种“测试驱动的开发”的方法在许多领域广泛应用，尤其是当我们需要写一些比较难写或者复杂的代码（比如我们这个例子）。因为在一个复杂度比较高的函数或者类中，你很难去定位一些小错误（比如语法错误，数学表达式出现了错误）。

✓

sigmoid 激活函数执行正确

正向传播

✓

隐藏层的输入在训练和运行函数中均正确执行

在Python 3 中， 新引入了一个运算符“@”， 可以用来进行矩阵乘法，你有兴趣可以看这里：
https://docs.python.org/3/whatsnew/3.5.html#whatsnew-pep-465

✓

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

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。

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。

这次的迭代次数比较合适，模型既没有过拟合，也没有欠拟合，loss 值也保持在一个较低的水平~
迭代次数的选择可以依据我们的训练和验证损失plot图来不断的调整。当我们的神经网络在验证集上过拟合的时候，验证集的损失开始增加并且训练集的误差应该保持下降趋势。一般考虑选择满足训练集损失足够小同时验证集损失没有升高附近的迭代次数。可以根据模型实际运行情况来修改，直到模型validation loss不再有明显下降。

✓

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。

我们对于模型复杂度的控制，主要体现在隐藏层单元的个数上，一般至少8个才能保证我们的模型不会欠拟合。你的选择是12个，模型表现的不错，很好。一般来说，隐藏单元越多，你需要学习的权重也就越多，这样需要的训练数据和训练时间也就越长。

✓

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。

学习速率的选择非常的关键，如果太小的话，就需要很多次的迭代才能够最终收敛，这样的话就需要耗费太多的时间训练模型。而如果学习速率太大的话，很可能出现无法收敛的情况。在你的例子中可以看到尽管training loss和validation loss有一些波动，但是training loss在保持下降，所以算是收敛的不错的了。
不同的模型，不同的神经网络结构（多少层，每层多少个units）都会影响我们的最佳学习速率。我们需要一开始设定一个大概的值，输出图形看效果后，根据图形判断是否收敛以及收敛速度后再对learning rate进行调整。

下载项目