

## 《机器学习》编程作业 4——神经网络

本次的训练营作业使用 `python+numpy` 实现。对于作业文件，推荐大家安装 `anaconda` 后使用 `jupyter notebook` 打开。

作业说明主要需要参考英文原版 `pdf` (在文件夹里)，本文件只是一个作业提示，告诉大家需要完成作业中的哪些地方以及注意事项。对于其他不需要大家完成的程序，我们鼓励大家看懂。

首先需要说明的是，每一个需要大家完成的点都会有如下类似的语句：

```
# your code here (appro ~ 1 lines)
```

大家需要把自己的代码写在这行注释下面，括号里是提示大家大概需要写几行代码。这里只是老师根据自己的经验对代码的估计，并不是说你一定要写这么多行。假如你写的代码多余注释中的估计行数，不要担心！只要你的代码能够正常运行就可以了。

在有些地方，为了降低编程难度同时提高代码规范性，我们已经为大家写了一部分代码，大家只需要修改即可。例如，在初始化参数 `theta` 中，你会看到如下的代码：

```
theta = None
```

这里，大家只需要删除赋值等式右边的 `None`，修改为自己编写的表达式即可。

本次作业需要完成的地方有：

- 1、`forward_propagate` 函数
- 2、`cost` 函数
- 3、`backprop` 函数

需要说明的是，这次作业使用了 `scipy.optimize` 库。`scipy.optimize` 是一个优化算法包。如果你没有听说过这个库，没有关系。正如吴恩达老师在视频中说的，你并不需要去手写自己的优化算法，也不需要看懂源码。

作业中为了导入数据，代码使用了 `scipy.io` 库。这是一个可以帮助我们导入 `.mat` 格式数据的包。

在完成一个地方的代码之后，我们强烈建议同学们运行一下代码看看是否报

错，同时通过对比英文原版作业中的期望输出值来检验自己的结果是否正确。

这是训练营的第四个作业，根据我们的经验，这个作业是所有编程作业当中难度最高的作业。如果你发现自己为了完成这次作业花费了很长的时间，这是很正常的事情。实际上，实现神经网络的反向传播的代码比较复杂。祝大家成功。