寒假关于《Deep Learning with Python》 总结

——任何一个小知识点应该在2-3天内完成，如果没有完成，说明你没有付出足够精力。

利用寒假基本读完了这本书（除去RNN部分），总体而言，这本书从基础的ML概念开始讲，并且结合Python的Keras库，适合新手入门。

现对寒假的成果进行总结。

在完成本书的过程中，使用keras的基本DL和CNN完成了kaggle里的digit recognition 和CNN实现猫狗分类（利用部分数据）。

学会了如何利用panda读取数据和保存数据，如何在kaggle上利用kernel提交数据。

知道了tensor的概念。

DL的本质就是去entangle data 去自动地学习feature，但是目前hyperparameters 如何优化还需要寻找更好地方法让其自动来。

学会了用keras做DL的流程：

1. Define problem. 读取数据，并作数据预处理，这一步主要包括将数据转化为tensor, 如果是多分类的将label转化为vector。预处理还包括normalization，这一点指如果多个feature的range不同时，将其化为同一range（小）；将数据转化到0-1之间，有利于weight的更新。(input data, what type of problem, 是否适合用ML, pick a measure of success, evaluation protocal)
2. 搭建网络，在搭建网络的过程中，可以使用已有的Sequential()也可以利用function的特点自定义model. 搭建网络主要是关注有几层，每一层的node个数，激活函数类型。但是最后一层往往是由题目类型决定的，regression不需要激活，因为是做预测，而classification分类问题一般是用sigmoid/softmax来针对2分和多分问题。
3. Compile(loss function, optimizer 和metrics) optimize目前比较常用的就是RMSprop, loss function和metrics也是由问题决定。
4. Fit－这一步主要要定好训练数据，label, validation, batch\_size，epoch的个数,，可以利用callback实现early stopping,. History里面存着中间的数据。直到overfit
5. Improve: 解决overfit问题，regularization(减小weight)，dropout(随机删掉node), 增加training data, reduce the size of network.
6. Predict

几个问题：

Validation 的作用：检验是否overfitting，为什么不用test因为会造成information leak.

K-fold 就是挨着作为validation，将loss求平均来作为衡量标准。在数据比较少的时候可以用。

机器学习的几大流派：

1. Probabilistic Modeling (Naïve Bayes) –eg. Logistic regression
2. Early Neural Network
3. Kernel method(SVM) find the boundary,不适合大的数据集，不适合感知问题
4. Decision tree, random forest and gradient boosting method

CNN

Filter

知道了目前几种比较流行的network,

VGG16

Inception

Res

知道了如何解决多输入／输出／graph

利用function功能

知道了如何做优化

use a pre-trained model -> fine tune

两种思路： 1. Feature extraction 2.fine tune

1. 直接使用已知数据，作为最后fully-connected 的输入，freeze前面的所有convolution层
2. 只freeze 前面几层，后面几层convolution也重新训练（所以花时间更长）

Model ensemble

Pooling together the predictions of a set of different models.

Key: diversity，不同的模型侧重点不一样，weighted average

通常是用tree-based methods and deep neural network

如何可以构建更深的层次

想要构建更深的层次要解决两个问题：gradient vanishing, bottleneck(某一层的近似中断导致信息无法向下传递)

跳一个（res）考虑之前的node

加入batch\_normalization(Inception)-所有神经层具有相同范围并且零均值的响应。

下一步：

结合kaggle学习tree的相关知识

－进而利用ensemble