Rossmann 销售预测报告

问题的定义

这是 Udacity 机器学习课程的毕业项目，项目中将应用神经网络求解销售预测问题。

销售预测

作为销售计划的中心任务之一，销售预测是指通过一定的手段和方法，对未来特定时间内全部产品或者特定产品的销售数量或销售金额的估计。在操作层面上，销售预测是在充分考虑了未来各种影响因素的基础上，结合本企业的过往销售业绩，运用一定的分析方法提出切实可行的销售目标。

影响销售的因素很多，包括有需求变化、经济变动等成分的外界因素，和有营销策略、生产状况、销售人员等成分的内部因素，但销售预测对于完善客户需求管理、指导运营、优化供应链、提高企业利润方面都具有重大促进作用，降低企业的业务计划的不确定性，因此，提出合理的销售预测一直也是人们孜孜不倦的追求，是企业辅助决策的重要工具。销售预测也几乎成为了所有商业智能（BI）的终极目标。

多年来，人们已经形成了定性分析法和定量分析法两类分析方法。其中，定量分析法中的趋势预测分析法和因果预测分析法在实际应用中也能取得一定的预测效果。他们可以基于历史数据，运用一些直观的算法，如算术平均法、指数平滑法，来进行预测，相比定性分析法，预测效果也有了一定的提高[1]。

但随着经济全球化，商业网络化的进程，市场竞争日趋激烈，业务复杂化，数据规模海量化，传统的预测方法已经越来越力不从心。预测的精确性成为销售预测的核心痛点，人们迫切希望一些性能更高，更智能的预测方法。数据挖掘[2]技术由计算机自动从大量数据集中提取隐含的、事先不知道但有潜在应用价值的信息，可用于学习复杂销售的复杂特征对于销售的影响，从而得到较好的预测效果。本课题将应用机器学习算法来实施销售预测。

问题分析

Rossmann 是欧洲的一家连锁药店，在欧洲 7 个国家拥有超过 3000 家药店。这是一个 Kaggle 比赛项目，本课题将按照项目中的说明，需要为 Rossmann 在德国的 1115 家药店做出提前 6 个星期的每日销售预测。

对于 Rossmann 的销售预测问题将是一个具体领域的销售预测问题，作为药店连锁店，具有一定的行业特征，这些将体现为数据特征。如上节所述，基于机器学习算法的数据挖掘技术会是一种可能的得到一个合适预测模型的办法。

本课题得到的预测模型将用于输出未来 6 个星期里每天的销售量，预测结果可以和实际销售情况对比，从而衡量预测效果。在实际使用中，还可以随着时间的推移，不断学习和预测。

评价指标

本项目将使用 Kaggle 项目中的评估函数 RMSPE（Root Mean Square Percentage Error）作为评估指标：

$$RMSPE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{y}_i}{y_i}\right)^2}$$

其中，y_i为一个门店某天的销售量，$\hat{y}_i$为相应的预测值。

RMSPE 评价指标特征分析：

1. RESPE：是误差的百分比，可以认为度量的是误差值对原始值的相对影响力。也就是说，如果原始值较大，对误差值相对宽容，如果原始值较小，对误差值相对严格。
2. RESPE 的取值范围为[0,1]

而其他常见评价指标，如 RMSE，度量的是“误差值”本身，相对而言，RMSPE 更适合作为销售额这样的数值较大的数据的评价指标。

研究分析

数据特征

作为一个 Kaggle 比赛项目，Kaggle 提供了项目数据，包括训练数据 train.csv，测试数据 test.csv，已经商店信息数据 store.csv。训练集数据的全部字段说明参见：https://www.kaggle.com/c/rossmann-store-sales/data，包括两部分：1，与销售日期相关的日期、促销、节假日特征值，2，回归值：销售额和顾客数量。测试数据并不包含回归值，只适合 Kaggle 在线测试，本项目采用离线实验的方式，故不使用。商品信息数据 store.csv 包括与每个门店相关的门店类别、促销情况、竞争门店信息。

另外考虑了，竞争门店的作用，并试图通过 *CompetitionOpenSinceYear* 和 *CompetitionOpenSinceMonth* 得到当日门店是否存在竞争门店，以及 CompetitionDistance 得到的竞争门店的距离，report2.html 就是以 CompetitionDistance 训练的模型，在测试集上的 RMSPE 错误率约为 12.6%，仍然有一定的改进空间。

为了获得更简洁的有效特征值，后来在 Kaggle 上面发现了门店的 State 属性数据集，通过引入 State 特征值，使用 Entity Embedding 算法的神经网络，得到了较好的预测效果。

另外，在 Store ID 确定的情况下，门店类别固定，不再必要；week 和 state 已知的情况下，StateHoliday 和 SchoolHoliday 也确定，不再必要。

总结：通过以上的分析和实际测试，所以本项目最终选取了如下特征值用于销售预测：

- Store：商店编号
- Date：数据统计日期
- Open：当天门店是否开放，节假日商品可能不开放。
- Promo：当天门店是否进行促销活动
- DayOfWeek：一周的星期几编号
- State：门店所属的地区，德国各地区的节假日各不相同，所以可以认为 State 也表达了 StateHoliday 和 SchoolHoliday 的属性。

而回归值，即要预测的数据为：

- Sales：当天销售额

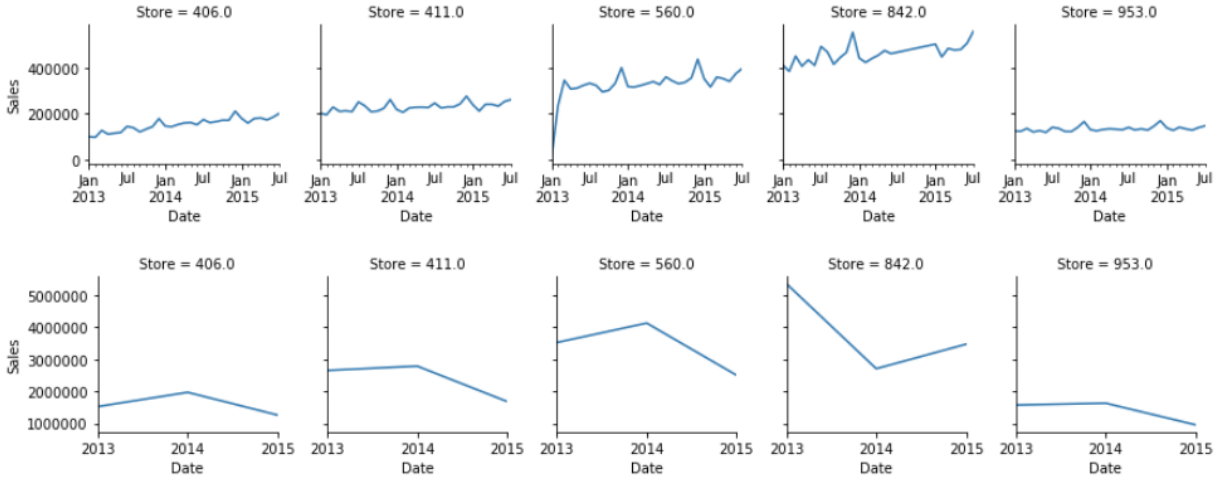
本项目将以离线的方式训练模型，并作出预测，以比赛项目的训练数据作为项目数据，分割出部分（0.5%）作为测试数据。

数据探索

本项目做了以下几种数据可视化探索：

1， 按月、按年对比：

随机选取 5 个 store 的销售总额趋势图：



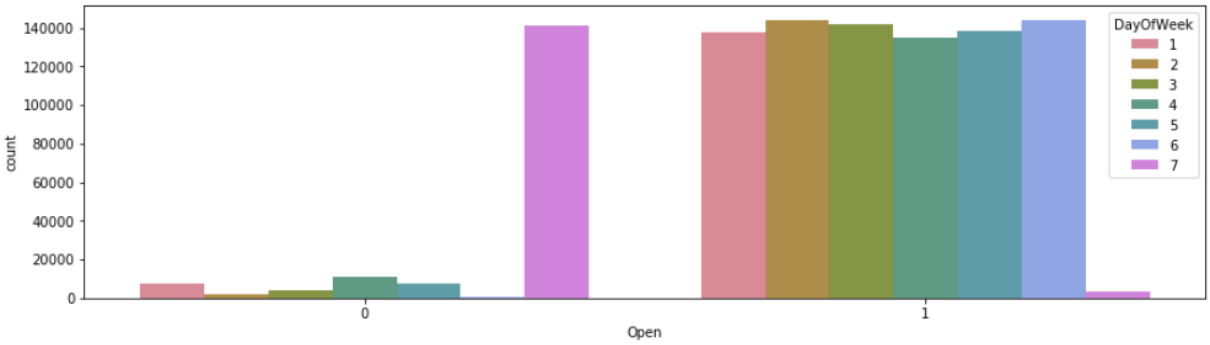
随机选取 5 个 store 的销售平均额趋势图：

时间上的可视化总结：

1， 各个门店销售趋势走势不尽相同

2， 总体销售额呈现上升趋势

2， Open 与销售额

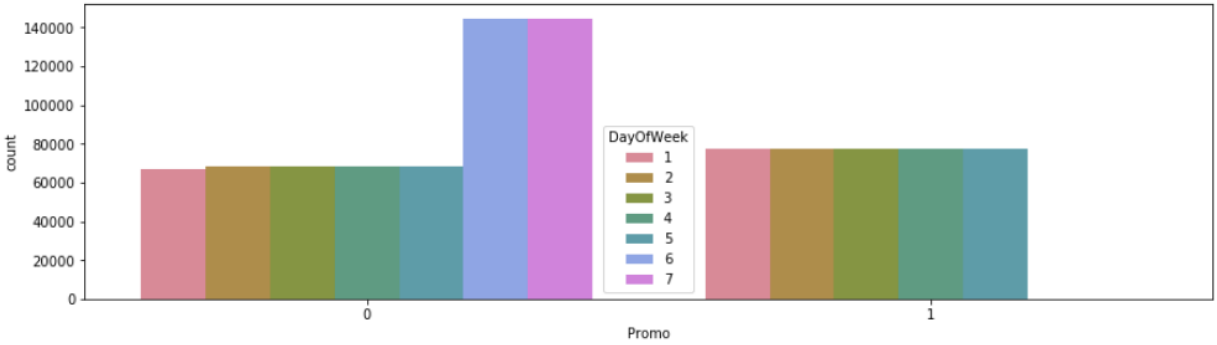


Open 特征值的可视化总结：

1， Open 对销售额有明显的区分作用，当 open=0 时，销售额几乎为 0，没有很好的区分度

2， Open 对销售额的影响是显著的，但却是确定的。

3， Promo 与销售额



Promo 与销售额的可视化总结：

1. Promo 对销售额有明显的区分作用：工作日保持了一定的销量的增长和稳定，周末一般没有 promo，销售额较大。
2. Promo 对销售额有一定的区分作用，结合 day of week 可以很好地描述销售额

算法分析

与传统预测方法相比，神经网络更擅长描述具有较强非线性、难以用精确的数学模型表达的复杂系统，并且具有一定的自适应能力。目前，神经网络模型已经成功地应用于许多领域，如经济预测、财务分析、贷款抵押评估等领域。鉴于（深度）神经网络的优势特点，结合本项目的以下特征和要求，所以本项目将选择使用神经网络，而不是 xgboost 等算法。

- 足够的训练数据，训练集上已经有 84 万+数据
- 销售数据希望结果越精确越好，也就是模型的准确度
- 性能要求不太严格，相对来说，模型准确度更重要

本项目研究了 Entity Embedding 算法，通过实验表明，变量压缩的做法不仅减少了内存的使用，而且相比 one-hot encoding 提高了算法的效率。

本项目采用 Keras 框架来实现具有 Entity Embedding 功能的神经网络。Keras 是一个 Python 的深度学习库，提供了高层次的 API，以便方便快捷地构建（深度）神经网络。

基准模型

本项目使用了两种 Linear Regression（线性回归）作为基准模型，详细结果参考 report.html “3. 基准模型”：

1， 总体线性回归
总体线性回归，是试图在整个数据集上训练模型，特征值选择了与 Embedding 网络模型同样的特征值。测试中取得的 RMSPE 的错误率不是很理想，只有 45%左右。

2， 基于门店的线性回归

考虑到，各个门店的差异较大，又尝试了基于门店的线性回归模型，即对每一个门店训练一个模型，测试取得的 RMSPE 错误率降到了 16%左右。

研究方法

数据处理

数据处理通过 prepare_data.py 实现，主要包含以下内容：

- 销售数据按日期排序，原始数据为时间降序排列，处理为升序排列
- 数据聚合：将 State 信息整合到 Store 信息中
- NAN 处理：将 nan 数据用“0”填充
- 数据序列化：将原始 3 个数据集序列化为 train_data.pickle 和 store_data.pickle。

- LabelEncoder：对选取的特征值进行 LabelEncoder 编码，并保持编码方案到 les.pickle 一边解码使用。Date 也拆分为 year，month，day 三项。
- 类型一致化：将编码后的特征值一致化为 int 类型。
- 数据标准化：在训练 Keras 模型的时候，对回归值 Sales 进行了 log 百分比的标准化，减少了数据稀疏性。

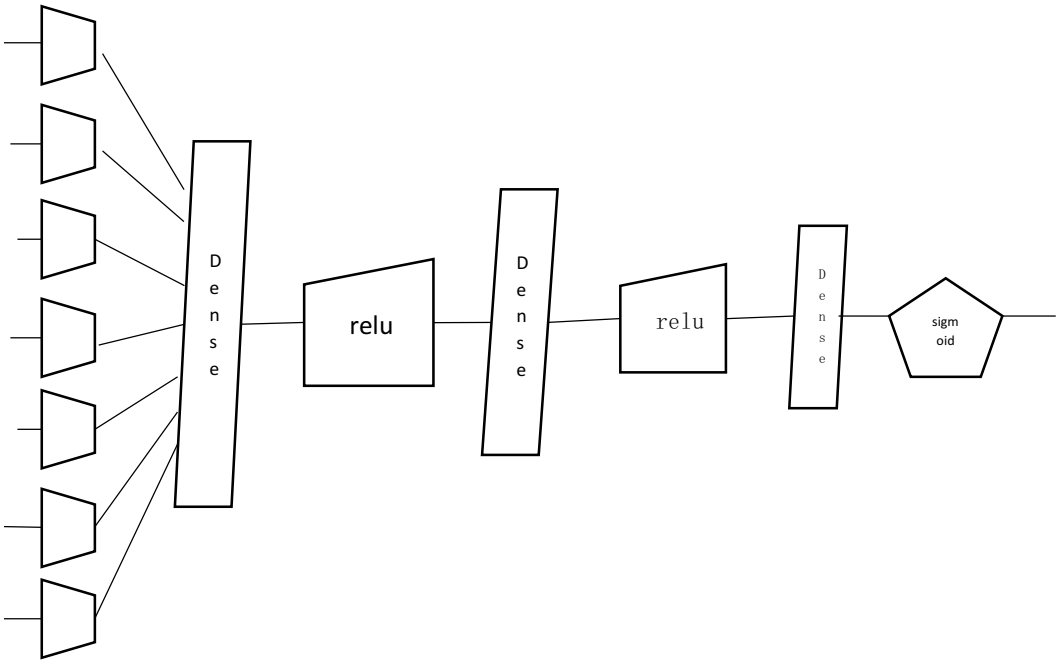
全部过程通过 prepare_data.py 的 prepare_data()实现。

算法实现

Entity Embedding 神经网络由 network.py 实现，项目中使用 Keras 框架构建神经网络，具体做法为：

- 特征值：模型训练使用了 Store ID，Day of week，promo，Date，Stat 组成的 7 项特征值，其中 Date 由 Year，Month，Day 三个子项组成。
- 网络模型：Sequential
- 网络层：
  - 每个特征值应用一个 Embedding 层
  - 全连接层：1000 维输出
  - 激活函数：relu
  - 全连接层：500 维输出
  - 激活函数：relu
  - 全连接层：1 维输出
  - 激活函数：sigmoid

网络层总体结构如图：



- Optimizer 和 Loss 函数

- Optimizer = adam
- Loss = mean_absolute_error

算法改进

本项目尝试了以下几种算法调整：

- **特征值的选取变换：从 CompetitionDistance 到 State：**

  就如同开题报告中提到了，首先尝试了使用 CompetitionDistance 作为一个特征值，具体做法为：

  1. 提取每天的 CompetitionDistance 值：由 Store 信息中的 CompetitionOpenSinceYear 和 CompetitionOpenSinceMonth 组成 CompetitionDate 和当前时间比较，如果当前时间小于 CompetitionDate，以 0 为 Distance，如果大于 CompetitionDate，则以 Store 信息中的 CompetitionDistance 为当天的 CompetitionDistance。
  2. 对 CompetitionDistance 编码：由第一步得到的 CompetitionDistance，定义一个距离类别变量：<=50 为 close，50 到 200 为 near，200 到 1000 为 normal，大于 1000 为 far（测试过程中进行了不同的调整）。最后再以其他特征值同样的 LabelEncoder 处理，并在 Entity Embedding 网络中用 Embedding 层处理。

  最后的测试结果为 report2.html 所示，测试集上的 RMSPE 误差为 0.126，仍然有改进的空间，所以找到了 State 数据集，并取得了更好的结果。

- **Optimizer 选取：从 sgd 到 adam**

  试验中，尝试了多种 Optimizer，如标准 sgd 和 adam，统计结果如下：

测试项目	sgd	adam
1 个模型	0.29	0.12
5 个模型	0.29	0.098

  所以，无论是一个还是多个模型，adam 的效果都比 sgd 要好。

- **网络层选择：**

  在全连接层的选择上，也做出了一些调整与尝试，如开始使用了 1 个全连接层，得到的最好的测试结果仅为 0.11，而选择 2 个全连接层时，可以达到 0.098。再次尝试添加全连接层（3 层）的时候，测试结果并没有明显改善，故最终使用了 2 个全连接层。

- **训练多个模型取均值**

  试验中，如果每次只训练一个模型，并在测试集上做预测，所得到的结果并不是很理想，比如在采用 adam 作为 Optimizer 的时候，一个模型上得到的 RMSPE 误差大约为 0.12，并不是特别理想。但如果训练 5 个模型，分别训练模型，并对各个模型的预测结果取平均值，所得到的 RMSPE 误差约为 0.098，就能得到了更理想的结果。

结果总结

模型评估

在选取了算法实现中的各项参数后，模型在测试集上取得了 RMSPE 误差 0.098 的结果，预测效果较为理想。

以下是项目中相关参数的总结，进一步从理论上说明了模型的可靠性：

1， adam Optimizer[3]
adam 优化方法可以动态调整每个参数的学习率，在于经过偏置校正后，每一次迭代学习率都有个确定范围，使得参数比较平稳。确保了一定的鲁棒性和稳定性，为模型的稳健型提供了一定的保障。

2， 输入层完备性
由算法实现中可以看到，本模型为选取的 7 个特征值分别建立了 tensor，这也为模型的训练提供了更好的学习方向，另外采取了 Embedding 层，减少内存使用的同时，也提高了算法的效率，但是相关参数需要一定的调整时间。

基准模型比较

基准模型（Linear Regression）与 Entity Embedding 神经网络的结果总结如下：

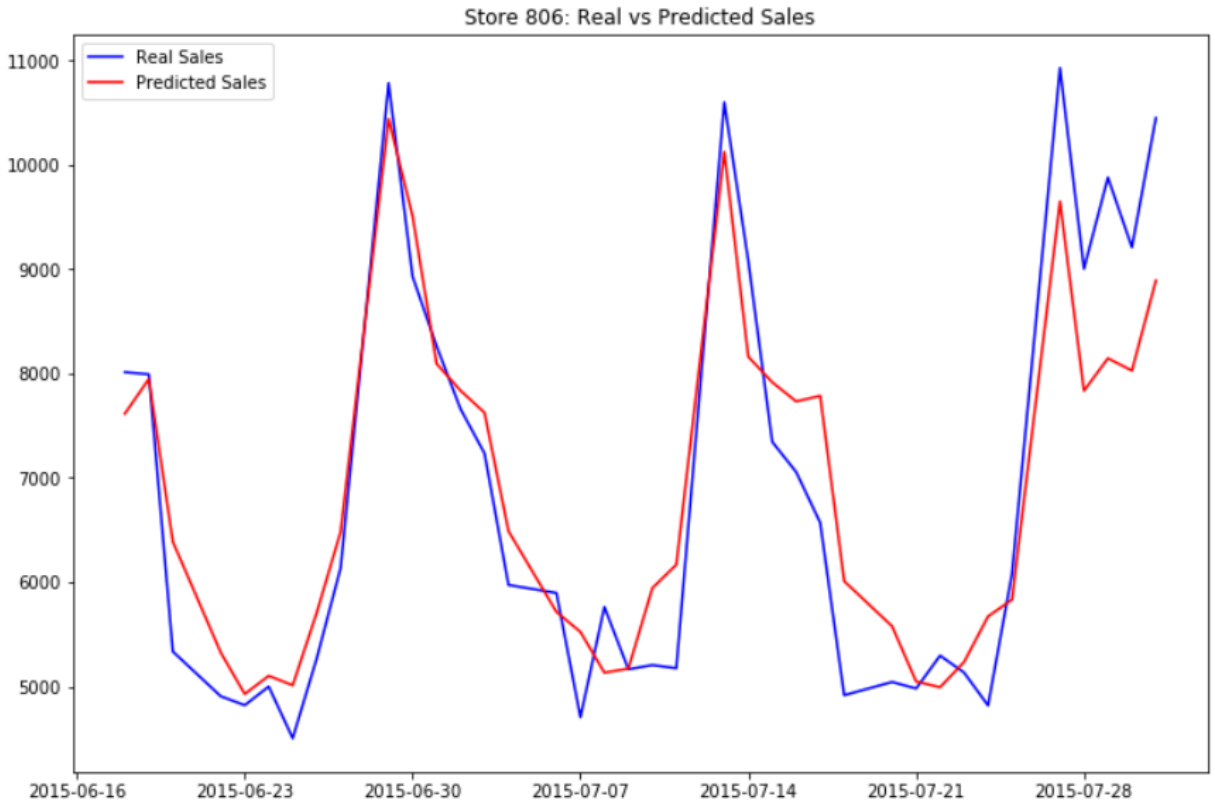
模型	总体线性回归	基于门店的线性回归	1 个模型神经网络	5 个模型神经网络
错误率	0.458	0.162	0.12	0.098

所以，本项目得到的模型，可以较好地拟合 Rossmann 的销售情况，预测效果也较为满意。

结论分析

预测结果可视化

report.html 中可以看到，在测试集中随机选取一个门店执行销售预测，并与真实数据一起做可视化分析统计。如门店 806 的 2015-06-16 到 2015-07-28 区间内的销售预测情况如下：



项目思考

本项目在实现过程中有如下两点思考：

1， 使用 Embedding 网络

根据本项目的特点，各个特征值均为，或者可以转换为，分类变量，适合 Embedding 网络。

由测试可知，Embedding 网络的效率较高，而且只需要选择相对简单的特征值就能取得较好的效果。由参考文献 1 可知，Entity Embedding 避免使用 one-hot encoding，由于 one-hot encoding 将占用更大的空间，使得神经网络变得更复杂。而 Entity Embedding 压缩了类型的维度，也达到类似编码的效果，是个有趣的处理分类特征值的方法，值得进一步研究。

2， 可视化的挑战与魅力

本项目开始阶段就试图对数据进行一些可视化，report.html 中销售额按年、月趋势的可视化操作就是一个例子。

由于之前没有完整的 python 用于可视化的数据处理和可视化操作，花费了不少时间。遇到的难点有：1，如何做数据的 roll up？基于门店的按年、按月统计，最终通过 play_ground 的 notebook 不断尝试，总结了如何分组数据，如何执行 resample 操作，从而达到想要的效果。2，何种方式执行可配置的可视化？为了达到较理想的可视化效果，同时支持可配置操作，也需要不断查找资料和尝试达到。在对 Open 和 Promo 的可视化时，使用了 seaborn 库，seaborn 库是一个很方便的可视化类库。

但是，可视化实现以后，数据就可以通过直观的方式显示，从而快速了解数据特征，进而方便决策。可视化是个强大的工具。

项目改进

本项目在开题报告中就打算使用 AR 模型作为基准模型，相比而言，AR 模型会比 linear regression 对于时间序列的拟合有更好的效果。但是实验中通过 SARIMAX 尝试建立了模型，但是没有得到按天的预测数据，时间紧迫，所以只好放弃。

如果说有什么改进，现在想到的，也许就是实现 AR 模型的基准模型，但这也不是本项目算法的改进。

参考文献

本项目参考的主要文献有：

【1】Entity Embedding of Categorical Variables: https://arxiv.org/abs/1604.06737

【2】Rossmann Store State 辅助数据：https://www.kaggle.com/c/rossmann-store-sales/discussion/17048

【3】Optimizer 比较：https://zhuanlan.zhihu.com/p/22252270

【4】Keras 文档：https://faroit.github.io/keras-docs/1.2.2/

【5】xgboost 与神经网络比较：http://www.sohu.com/a/145769790_697750