基于 TensorFlow 机器学习框架的房价批量评估

杨振鹏

摘要：TensorFlow 是美国谷歌公司最初于 2015 年发布的一个开源机器学习系统。该框架集成了大量机器学习方面的算法，使得基于它开发应用系统较为高效。在数据量充裕，且对自动化要求较高的应用场景中，TensorFlow 人工神经网络可以作为机构进行房价批量评估的选择之一。本文给出了一个基于 TensorFlow 构建房价的人工神经网络模型，并用于批量评估的实例，可以为有需求的机构和研究者提供参考。

关键词：TensorFlow，机器学习，人工神经网络，房价，批量评估

一、引言

TensorFlow 是美国谷歌公司最初于 2015 年发布的一个开源机器学习系统，由于有着灵活易用、开源免费和性能强大等特点，近年来受到人工智能领域业界的广泛欢迎。特别是 2017 年后该工具功能不断升级，目前研究人员使用 TensorFlow 搭建的机器学习模型，即可以在高性能的分布式系统中进行高强度训练，其训练结果也可以在轻量级的移动端直接应用，十分适合掌握大数据的机构快速开发应用系统[1]。TensorFlow 和相关 Python 开源项目丰富的内置库，将应用程序的开发工作量大大降低，以本文为例，一段用 Excel CSV 数据训练 BP 神经网络房价评估模型的程序只需要 30 余行代码，基于该模型对新增数据执行批量评估的程序也只需要 10 余行代码，对房地产企业和估价机构来说易于应用。

从另一方面来说，近年来我国房地产行业的数据可得性有了大幅提升，政府、经纪机构、评估机构、银行等积累了大量真实的原始交易记录。对房价进行批量评估的需求也日益提升，房地产贷款审核、房地产税的税基评估、开发商新建楼盘“一房一价”定价、门户网站清理虚假房价信息，都需要批量评估的手段[2]。对于上述应用场景而言，采用 TensorFlow 机器学习框架可以作为一种高效的辅助工具，特别是数据量达到百万量级、房屋特征属性多达数十列之后，人工神经网络模型相对与传统模型的优势将会显现。

二、人工神经网络房价评估模型的优势

关于人工神经网络和机器学习的背景知识不在本文赘述，希望了解这一领域

的读者可以参阅参考文献[1]，这里简要分析人工神经网络模型相对于传统房价批量评估模型的优势。

（一）模型设置的容错性更好

笔者认为，人工神经网络模型在批量评估中应用的最大优势，就在于模型设置的容错性更好。传统模型在求参数估计值时，往往需要用到多元回归分析（MRA），但多元回归分析需要建立在很多假设成立的前提下才能得到正确结果。特别是房屋特征属性可能具有很强的自相关性，而房价作为被解释变量也经常存在异方差的问题，对计量经济学不熟悉的从业者可能会设置错误的模型形式，得到有严重问题的回归结果。

举例而言，假如按常见做法，在数据中给每个小区都设置一个哑元变量（“主体变量”），用于提高评估模型针对不同小区的拟合准确度[3]，如果这时再不慎引入一些数值型的小区层面的特征变量，如“绿化率”、“容积率”，那自变量之间就会出现严重的自相关，甚至导致做最小二乘回归时矩阵无法求逆，使自动程序中断。即便回归能够完成，特征变量相应的系数回归值也经常偏离合理区间。

因此，基于多元回归分析的房价批量评估系统，都应编写一整套数据预处理、假设检验、反复调整模型中的解释变量的程序，否则就仍然要依靠有计量经济学知识和经验的人员“手工”进行模型设置，达不到“自动”批量评估的目标。

而人工神经网络模型由于采用反向传播（Back Propagation，BP）算法调整参数，对解释变量筛选的容错性更好。实践中可以把所有收集到的房屋特征属性全数通过标准化的变换输入神经网络，由神经网络自行学习其中规律，减少人为设置引发的偏误。

（二）对非线性关系拟合更好

正如大多数文献提到的，多层人工神经网络对非线性关系的拟合效果更好，而且不必指定非线性关系的具体形式，由神经网络自行适应[4]。另一方面，有的字典型房屋特征属性需要交叉组合，才能体现对房价的影响。举例而言，同一个朝向的房屋，在不同小区的受欢迎程度可能是不同的——如在甲小区南侧有特高压走廊，朝南的户型价格偏低，而在其他小区可能正好相反。这种属性的交叉组合效应，在传统模型中一般不会考虑，而在多层人工神经网络中会得到一定体现。因此，从实践效果来看，人工神经网络批量评估的各项效果评价指标（外推预测

的 MAPE、扩散率等）经常略好于基于同一个数据集的 MRA 模型。

（三）算法发展迅速，性能大幅提高

近年来，机器学习领域的算法发展迅速，特别是在处理大数据方面，性能大幅提升。当样本量超过百万数量级，房屋特征属性多达数十列之后，人工神经网络模型相对与传统模型的优势将会显现。特别是房屋特征属性中含有大量字典型变量时（如小区名称、房屋朝向、户型），添加哑元变量组后，造成矩阵阶数过高，求矩阵逆时计算量巨大。在这种情形下，采用 TensorFlow 机器学习框架中已集成的优化算法会大大提高模型的训练效率。

三、人工神经网络房价评估模型的不足

（一）评估过程难于解读

人工神经网络模型相对于传统评估方法也有劣势，其中最明显的一点就是它几乎将评估过程封装为一个黑箱，让人很难解读系统为什么对特定的房屋输出高价或低价。这在有些情形下不能被委托人接受。

（二）人工神经网络模型的经济学意义不明确

人工神经网络模型的经济学意义也有待明确，使其只能作为一个唯象意义上的工具，而忽略了房地产市场背后经济学原理。

（三）人工神经网络模型存在过度学习的可能性

人工神经网络模型对样本内拟合的能力很强，但在训练达到一定规模后，有可能出现过度学习。形象地比喻，就是过分关注一些仅对样本才成立的偶然性规律，过多学习这些偶然规律后，在普遍的外推预测中反而表现更差。因此，一般需要预留一部分样本作为测试是否过度学习的验证样本。在小数量的情形下，这不利于充分利用已有数据，得到的模型也可能不如传统模型准确。

尽管存在这些不足，在对“解释过程”要求不高而数据量充裕的应用场景中，人工神经网络还是可以作为机构进行房价批量评估的选择之一。

四、基于 TensorFlow 实现 BP 神经网络房价评估的实例

（一）数据集

这个演示用于评估某一地区 101 套住房在 2011 年 3 月 15 日的价格，为了获取准确的评估模型，收集了该地区在 2011 年 3 月成交的 302 宗可比交易案例。数据概貌如下表所示。

表 1 待评估房屋和可比案例数据集的特征统计

待评估房屋				可比案例			
样本量	101	涉及小区	46	样本量	302	涉及小区	46
平均总价	275.69 万	平均面积	102.94m^2	平均总价	234.72 万	平均面积	91.49m^2
户型种类	8	朝向种类	10	户型种类	8	朝向种类	10
平均楼层	9.3	平均总层	17.9	平均楼层	9.2	平均总层	16.8

上述可比案例数据收集在 Sample.csv 文件中，待评估房屋数据收集在 MA.csv 文件中。原始数据可在 http://www.fangtax.com 上下载。

（二）在 TensorFlow 中建立并训练模型

本文程序的运行环境为 Python 3.5.2、TensorFlow-GPU 1.8、numpy 1.14.5、pandas 0.23.3。上述环境均为开源项目，可在各自的官方网站上免费下载使用。准备好测试数据 Sample.csv 后，创建一个 Python 脚本，程序起始部分先导入需要用到的 tensorflow、numpy、pandas 库：

```
1  import tensorflow as tf
2  import numpy as np
3  import pandas as pd
```

接下来用一个字符串列表 scalarVars 记录哪些房屋特征属性字段是“数值型”字段，用 dictVars 记录哪些是“字典型”字段：

```
4  scalarVars = ['地上总层','所在楼层','面积','物业费','容积率','建筑年代','地铁房','绿化率']
5  dictVars = ['行政区','环线位置','朝向']
6  dictValues = {}
```

然后调用 pandas 的内置功能，读出 Excel CSV 文件中的全部数据：

```
7  df = pd.read_csv('Sample.csv',index_col='ID')
```

这时 df 是一个 DataFrame 类型的数据表，包含了 CSV 文件中所有字段的数据。接下来，先提取出刚才设定的 8 个数值型字段，并利用 DataFrame 内置的平均值 mean()、标准差 std()函数，对所有数值型字段进行归一化。受益于这些内置功能，这步工作仅需 2 行代码：

```
8  dfScalar = df[scalarVars]                              #先分离出数值型字段
9  dfX = (dfScalar - dfScalar.mean())/dfScalar.std()      #对所有数值型字段进行归一化
```

同样，利用 pandas 可自动添加哑元字段的功能，逐一为每个字典型房屋特征字段创建哑元字段组 dfTemp；并利用数据表的连接（concat）功能，将 dfTemp 并入刚才存放归一化数值型字段的数据表 dfX：

```
10 for catName in dictVars:
11     dfTemp = pd.get_dummies(df[catName])     #利用 pandas 内置功能创建哑元字段组
12     dictValues[catName] = dfTemp.columns.values.tolist()[1:]      #保存字典的取值列表
```

```
13      dfX = pd.concat([dfX,dfTemp.iloc[:,1:]],axis=1)   #将创建的哑元字段组并入 dfX 中
```

由于 pandas 为每个字典取值都添加了 1 列哑元字段，因此从信息的角度来讲，总有 1 列是冗余信息。故程序第 13 行执行合并时，对 dfTemp 用了 iloc[:,1:] 操作进行切片，作用是取数据表的所有行，但忽略第 1 列；axis=1 的含义是前后两张数据表按增补“列”的方式进行合并。

以上程序执行后，dfX 就是经过预处理后的所有房屋特征的数据表，即自变量。为了将其用于后续的人工神经网络模型训练，还要把它转为 numpy 的矩阵（array）类型对象 dt_x；同时，对于被研究的因变量“总价”，这里进行一个求自然对数的变换，并也将数据调整成 N×1 阶的矩阵 dt_y：

```
14  dt_x = np.array(dfX)                                  #将自变量数据集 dfX 转为矩阵
15  dt_y = np.log(np.array(df['总价']).reshape(df.iloc[:,0].size,1))    #生成因变量 log(总价)
```

第 15 行代码中的 df.iloc[:,0].size 是获取总样本量 N（即 df 数据表的总行数），之后利用 numpy 矩阵的 reshape 功能调整 dt_y 为 N×1 矩阵。

上述程序即完成了数据的读取和预处理。

之后，第 16-23 行代码建立神经网络的结构。本文采用如下包括 1 个隐含层的神经网络结构：

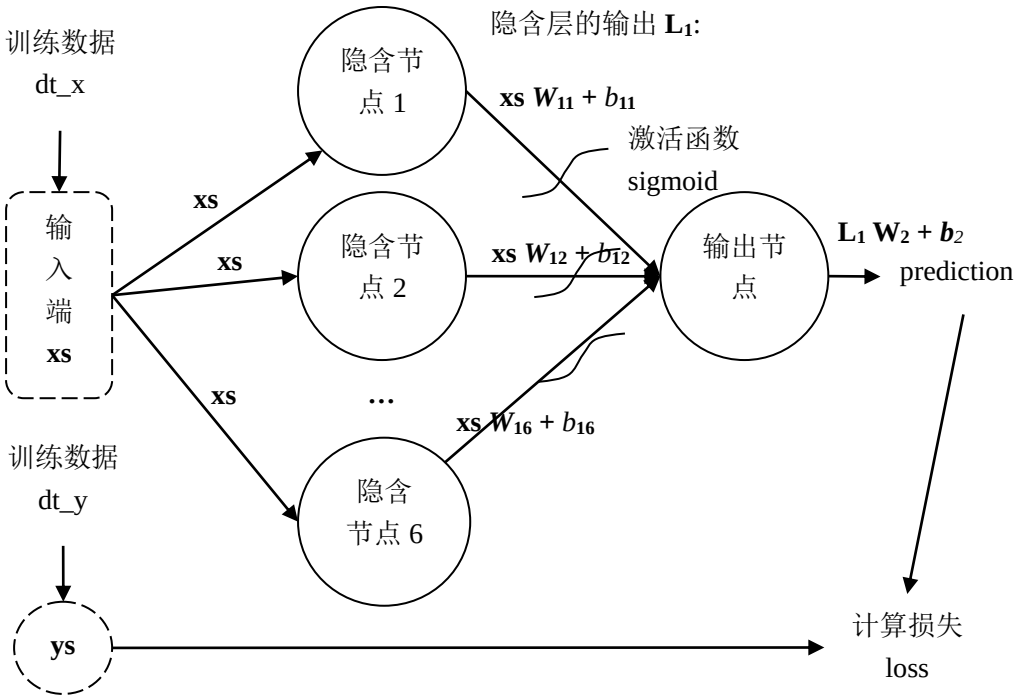

图 1 本文采用的人工神经网络结构图

在 TensorFlow 中构建网络的程序思路是：①先利用 TensorFlow 的占位符

placeholder 函数，建立数据输入端的神经节点 xs、ys，其中 xs 是 n×M 阶矩阵（行数 n 设置为“None”是因为将来输入数据的样本量 N 是变值，M 是自变量矩阵的列数，即自变量数目 M = dfX.columns.size），ys 是 n×1 阶矩阵；②创建隐含层的可训练参数 W1、b1，其中 W1 是 M×6 阶矩阵，b1 是 6 个元素的数组；③定义隐含层的输出 L1，即在矩阵运算 xs W1 + b1 的基础上，采用激活函数 sigmoid 调整输出（感兴趣的读者可尝试其他 TensorFlow 内置的激活函数）；④照相同的步骤定义输出层，本文输出层没有设置激活函数。整段代码如下：

```
16  xs=tf.placeholder(tf.float32,[None,dfX.columns.size])   #定义神经网络的输入节点 X
17  ys=tf.placeholder(tf.float32,[None,1])                  #定义神经网络的输入节点 Y
18  W1=tf.Variable(tf.random_normal([dfX.columns.size,6]))          #定义隐含层
19  b1 = tf.Variable(tf.zeros([6]) + 0.1)
20  L1 = tf.nn.sigmoid(tf.matmul(xs, W1) + b1)
21  W2 = tf.Variable(tf.random_normal([6, 1]))                      #定义输出层
22  b2 = tf.Variable(tf.zeros([1]) + 0.1)
23  prediction = tf.matmul(L1,W2)+b2
```

之后利用 TensorFlow 内置的算法，设置损失函数、训练算法和初始化函数。损失函数 loss 为人工神经网络输出值 prediction 和实际值 ys 的平均平方误差；训练算法采用 TensorFlow 内置的 Adadelta 梯度下降反向传播算法[5]，最小化 loss，实现神经网络参数随数据训练的自动调整；初始化函数采用默认设置：

```
24  loss=tf.reduce_mean(tf.reduce_sum(tf.square(ys-prediction),reduction_indices=[1]))
25  train_step=tf.train. AdadeltaOptimizer(0.2).minimize(loss)      #设置反向传播算法
26  init=tf.global_variables_initializer()                          #设置初始化函数
```

上述设置完成后，即可以创建一个 TensorFlow 会话线程，进行初始化，并基于前文预处理后的数据 dt_x 和 dt_y 对刚才创建的神经网络进行 50000 轮训练：

```
27  with tf.Session() as sess:                                      #打开 TensorFlow 会话
28      sess.run(init)                                              #初始化
29      for i in range(50000):                                      #迭代训练 50000 次
30          sess.run(train_step, feed_dict={xs: dt_x, ys: dt_y})    #使用 dt_x、dt_y 训练网络
31          if i % 100 == 0:                                        #输出损失衰减的情况
32              print(sess.run(loss, feed_dict={xs: dt_x, ys: dt_y}))
```

每 100 轮输出一次损失值，根据程序输出的损失衰减情况，决定是否增加迭代次数。至此已完成房价评估模型的建立和训练。

（三）利用训练好的人工神经网络执行批量评估

利用训练好的人工神经网络对 101 宗待评估案例执行批量评估，须先读入并预处理数据：

```
33      dMA = pd.read_csv('MA.csv',index_col='ID')
34      dXA = dMA[scalarVars]
35      dXA = (dXA - dfScalar.mean())/dfScalar.std()
36      dXD = dMA[dictVars]
37      for catName in dictVars:
38          for colName in dictValues[catName]:
39              dXD[colName] = dXD.apply(lambda r: 1.0 if r[catName]==colName else 0.0,
                axis=1)
40      dXA = pd.concat([dXA,dXD.iloc[:,len(dictVars):]],axis=1)
41      dt_xa = np.array(dXA)
```

之后，可以利用训练好的网络进行评估，并将结果保存在另一个 csv 文件中：

```
42      dt_ya = sess.run(prediction, feed_dict={xs:dt_xa})
43      dt_ya = np.exp(dt_ya).reshape(dt_xa.shape[0],1)
44      dfOut = pd.DataFrame(dt_ya, columns = ['TF 评估价格'])
45      dfOut.to_csv('res.csv', index=False, header=False)
```

用实际交易价格计算 IAAO 1999 对于批量评估的几项主要评价指标，并与传统 MRA 评估模型相比，对比如下表所示：

表 2 模型的评估效果评价指标

IAAO 1999 评价指标	理想值范围	MRA 模型		神经网络模型	
		结果	是否合格	结果	是否合格
算术平均评估水平	0.9 ~ 1.1	0.984	√	1.004	√
加权平均评估水平	0.9 ~ 1.1	0.979	√	1.000	√
中位数评估水平	0.9 ~ 1.1	0.974	√	1.009	√
PRD 价格相关差	0.98 ~ 1.03	1.005	√	1.004	√
COD 扩散系数	< 10	6.01	√	6.67	√

可以看到神经网络模型各项指标均符合要求，大部分指标略好于传统 MRA 模型的效果。但扩散系数指标略大于传统 MRA 模型，这一点值得后续研究。

五、结论

TensorFlow 框架是一种新出现的机器学习工具，由于集成了大量算法，基于该框架开发应用系统较为高效。在数据量充裕，且对自动化要求较高的应用场景中，TensorFlow 人工神经网络可以作为机构进行房价批量评估的选择之一。本文的实例可以为有这方面需求的机构和研究者提供参考。

参考文献：

[1] Sam Abrahams，Danijar Hafner，Eric Erwitt，Ariel Scarpinelli 著，段菲，陈澎译. 面向机器智能的 TensorFlow 实践[M]. 北京：机械工业出版社，2017

[2]杨振鹏，马亚男. 计算机辅助批量估价在房地产信息系统中的实现[J]. 中国房地产估价与经纪，2011(4)：24-31.

[3]刘洪玉，杨振鹏. 基于主体变量的住房价格批量评估[J]. 统计与决策，2012(3)：62-66.

[4]李菊. BP 神经网络在房地产批量评估中的应用研究[D]. 昆明理工大学，2015.

[5] Matthew Zeiler. Adadelta: An Adaptive Leaning Rate Method[EB/OL]. arXiv:1212.5701 v1 [cs.LG] 22 Dec 2012. http://matthewzeiler.com/wp-content/uploads/2017/07/googleTR2012.pdf

作者信息：

单位：北京城建兴华地产有限公司

地址：北京市海淀区丹棱街 16 号，邮编 100080

联系电话：18610193102

电子邮箱：1392033969@qq.com