

实 验 报 告

|  |  |
| --- | --- |
| **课程名称：** | 大数据分析基础训练 |
| **项目名称：** | 人民币对美元汇率的大数据分析 |
| **姓 名：** | 张茂桢 |
| **学 号：** | 2016034643044 |
| **班 级：** | 16信计 |
| **指导教师：** | 吴世枫 |

**数学与系统科学 学院 工业中心 实验室**

二〇〇 年 月 日

**广东技术师范学院实验报告**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 学院： | 数学与系统科学学院 | | | 专业： | | 信息与计算科学 | | | 班级： | | 16信计 | | 成绩： |  |
| 姓名： | 张茂桢 | | 学号： | | 2016034643044 | | | 组别： | |  | | 组员： |  | |
| 实验地点： | |  | | | 实验日期： | |  | | | | 指导教师签名： | | |  |

项目名称： 人民币对美元汇率大数据分析与预测

目录

[一、实验项目名称 5](#_Toc28529623)

[二．实验说明 5](#_Toc28529624)

[2.1 实验内容 5](#_Toc28529625)

[2.2实验知识点 5](#_Toc28529626)

[2.3 实验环境 5](#_Toc28529627)

[2.4 适合人群 5](#_Toc28529628)

[2.5 代码获取 6](#_Toc28529629)

[三．开发准备 6](#_Toc28529630)

[四．项目文件结构 8](#_Toc28529631)

[五、实验内容及步骤 8](#_Toc28529632)

[5.1导入所需要的库 8](#_Toc28529633)

[5.2数据预处理 8](#_Toc28529634)

[5.3设置基本参数 9](#_Toc28529635)

[5.4构建lstm模型主体 10](#_Toc28529636)

[六．实验数据记录和处理 11](#_Toc28529637)

[6.1模型保存 11](#_Toc28529638)

[6.2数据处理 11](#_Toc28529639)

[6.3训练数据 11](#_Toc28529640)

[七．实验结果与分析 12](#_Toc28529641)

[7.1预测测试样本 12](#_Toc28529642)

[7.3预测未来20天的值 14](#_Toc28529643)

[14](#_Toc28529644)

[15](#_Toc28529645)

[15](#_Toc28529646)

[八．问题与讨论 15](#_Toc28529647)

[8.1分析结果 15](#_Toc28529648)

[8.2不足与改进 16](#_Toc28529649)

[1、单变量输入 16](#_Toc28529650)

[2、新版本tensorflow不再支持。 16](#_Toc28529651)

# 一、实验项目名称

本实验研究的是人民币兑美元汇率的大数据分析与预测。

# 二．实验说明

## 2.1 实验内容

通过现有的数据，建立数学模型，预测出未来一个月的汇率变化趋势。本文采用美元对人民币的汇率数据，时间是1990年1月1日到2018月8月21日。利用Tensorflow搭建模型并完成预测，数学模型用Lstm（长短期记忆）人工神经网络，Lstm是RNN（循环神经网络）的一种变形，是在RNN的基础上施加了若干个门来控制，使其在长时间步的传递过程中减少信息失效的可能。

## 2.2实验知识点

Python基本知识

Python基础库如numpy，panda，matplotlib

TensorFlow模块的使用

Lstm神经网络

## 2.3 实验环境

运行环境window7

开发环境python3.6

调试环境jupyter notebook

## 2.4 适合人群

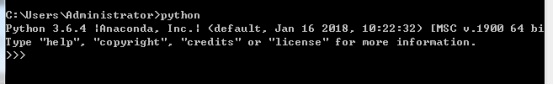
对机器学习有兴趣的在校学生或个人。

## 2.5 代码获取

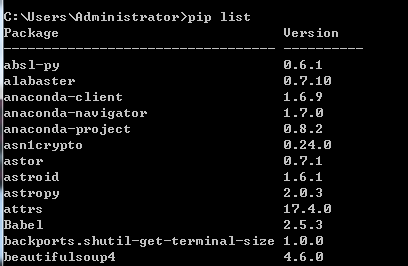
# 三．开发准备

Anaconda是一个方便的python包管理和环境管理软件，一般用来配置不同的项目环境。  
我们常常会遇到这样的情况，正在做的项目A和项目B分别基于python2和python3，而第电脑只能安装一个环境，这个时候Anaconda就派上了用场，它可以创建多个互不干扰的环境，分别运行不同版本的软件包，以达到兼容的目的。  
Anaconda通过管理工具包、开发环境、Python版本，大大简化了你的工作流程。不仅可以方便地安装、更新、卸载工具包，而且安装时能自动安装相应的依赖包，同时还能使用不同的虚拟环境隔离不同要求的项目。

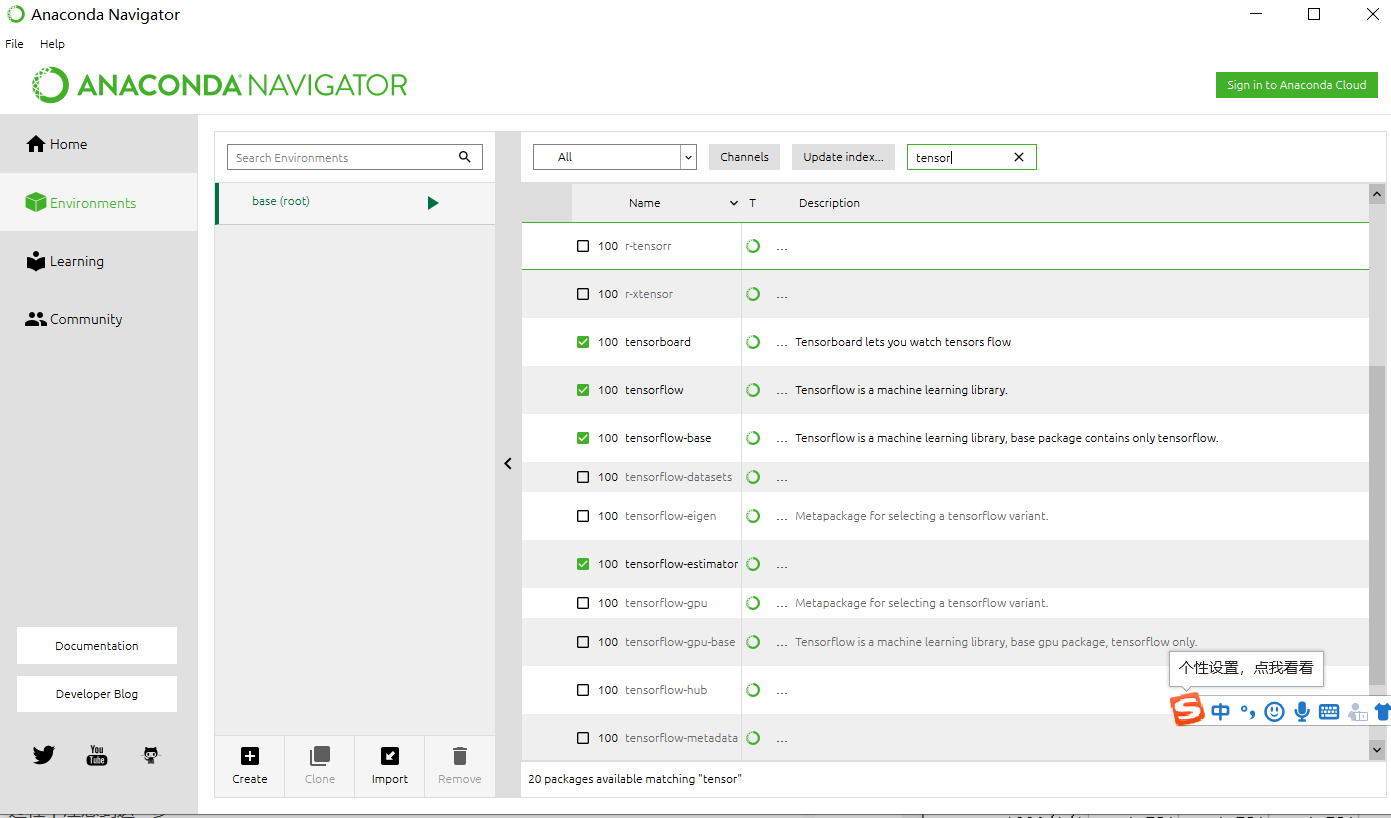




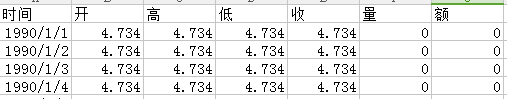
输入pip list查看库列表，部分截图如下



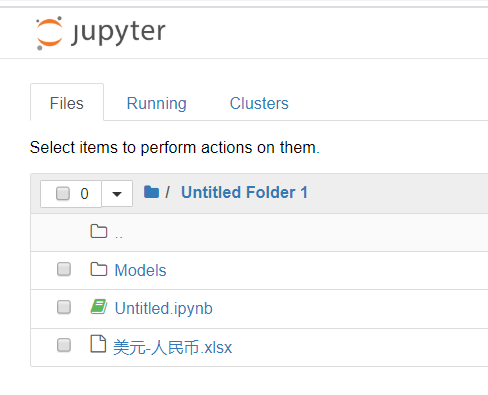
通过查看可以发现还需要安装TensorFlow库

实验所需文件部分数据如下所示



# 四．项目文件结构



# 五、实验内容及步骤

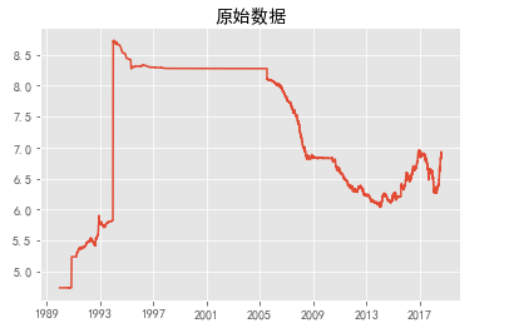
## 5.1导入所需要的库



## 5.2数据预处理



输出：



## 5.3设置基本参数



## 5.4构建lstm模型主体



# 六．实验数据记录和处理

## 6.1模型保存

|  |
| --- |
| '''初始化LSTM模型，并保存到工作目录下以方便进行增量学习'''  regressor = SKCompat(learn.Estimator(model\_fn=lstm\_model, model\_dir='Models/model\_1')) |

## 6.2数据处理

调用前面定义好的缩放函数处理原始数据，这里先用7000个数据作为训练样本，这7000个训练数据做处理将得到6989个数据，因为每12步取一组x，第13步取为y，假设希望剩余的447个数据在测试时都预测出来，则我们需要取第6989个数据到最后一个数据，用来转换为测试样本。

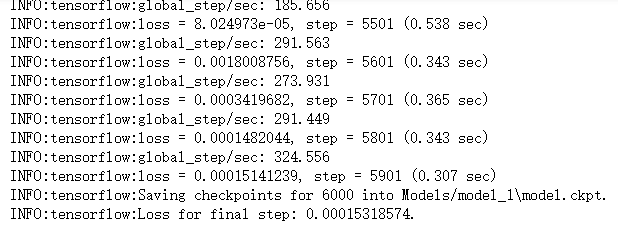
|  |
| --- |
| '''对原数据进行尺度缩放'''  data = data\_processing(data)  '''将7000个数据来作为训练样本'''  train\_X, train\_y = generate\_data(data[0:7000])  '''将剩余数据作为测试样本'''  test\_X, test\_y = generate\_data(data[6989:-1]) |

## 6.3训练数据

以仿sklearn的形式训练模型，这里指定了训练批尺寸和训练轮数。

|  |
| --- |
| regressor.fit(train\_X, train\_y, batch\_size=BATCH\_SIZE, steps=TRAINING\_STEPS) |

训练结果如下：

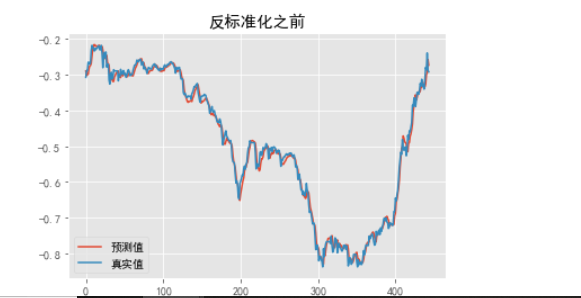


# 七．实验结果与分析

## 7.1预测测试样本

用训练好的模型来计算处理过的测试集，可以得到对应预测值，预测出来的数据为标准化之后的，并非原始数据，画出对比图如下。

|  |
| --- |
| '''利用已训练好的LSTM模型，来生成对应测试集的所有预测值'''  predicted = np.array([pred for pred in regressor.predict(test\_X)])  '''绘制反标准化之前的真实值与预测值对比图'''  plt.plot(predicted, label='预测值')  plt.plot(test\_y, label='真实值')  plt.title('反标准化之前')  plt.legend()  plt.show() |

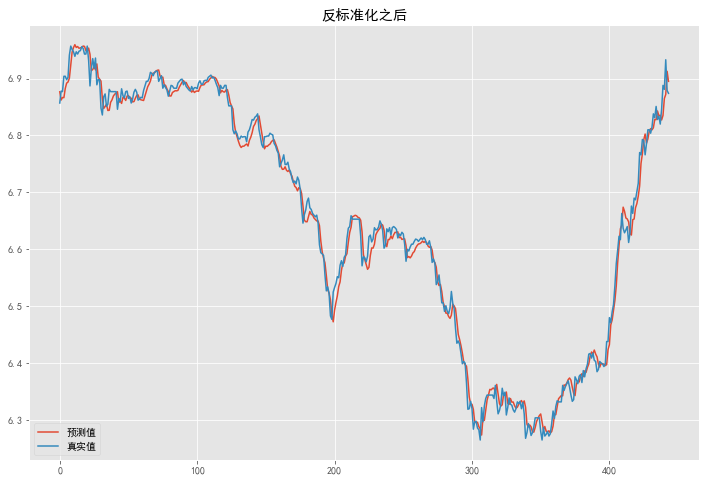


为了能看到真实数值，再定义一个反标准化函数。

|  |
| --- |
| '''自定义反标准化函数'''  def scale\_inv(raw\_data,scale=True):      '''读入原始数据并转为list'''      data = pd.read\_excel('美元-人民币.xlsx',header = 0, sheetname='Sheet1')      data = data.iloc[:, 4].tolist()      if scale == True:          return raw\_data\*np.std(data)+np.mean(data)      else:          return raw\_data\*(np.max(data)-np.min(data))+np.min(data) |

调用反标准化函数即可得到原始值，并画图观察。

|  |
| --- |
| sp = scale\_inv(predicted)  sy = scale\_inv(test\_y)  '''绘制反标准化之后的真实值与预测值对比图'''  plt.figure(figsize=(12,8))  plt.plot(sp, label='预测值')  plt.plot(sy, label='真实值')  plt.title('反标准化之后')  plt.legend()  plt.show() |



## 7.3预测未来20天的值

按照前面的思路，可以使用12个数据预测后一天的值，如果需要预测20天的值，可以用前面已知的12天数据预测下一天的值，这个值又放入数据中作为已知值来继续预测下一天的值，打印出预测结果。

|  |
| --- |
| day=20  l=len(data)  for i in range(day):      P=[]      P.append([data[l-TIMESTEPS-1+i:l-1+i]])      P=np.array(P, dtype=np.float32)      pre=regressor.predict(P)      data=np.append(data,pre)  pre=data[len(data)-day:len(data)+1]  print(pre) |

## 

同样这个值是标准化的值，可以打印出反标准化的值和画出预测图。

|  |
| --- |
| #反标准化的值  print(scale\_inv(pre))  #预测图  p = plt.figure()  plt.plot(scale\_inv(pre))  plt.show() |

## 

## 

# 八．问题与讨论

## 8.1分析结果

用训练好的模型，应用在测试集时，得到的预测结果还是令人满意的，预测出的数据误差都是小于0.1的，在0.05的误差范围内，准确率也达到97%。

对于预测未来20天的值，从预测图可以看到，数值呈下降的趋势且下降幅度较快，预测的结果其实是不准确的，原因也比较明显，在前面的测试集中，虽然是预测447个数据，但是每一次的预测都是用已知的真实值来计算的，也就是说，预测出来的值并不会加入模型中继续训练，也不会用预测出来的值作为输入参数。这预测的447个需要预测的值，本身已经在输入中了。

预测未来20天值，第一个还算是比较准确的，按照前面的做过的测试，准确率应该是达到百分九十七的，后面的数据预测时，输入是用预测出来的数据补充的，所以可能会不太准确。

## 8.2不足与改进

### ****1、单变量输入****

在本文中，只取收盘价作为输入，在实际中，也是不合理的，因为影响汇率的因素有很多，可以尝试加入其他变量，预测结果可能会更好。

### ****2、新版本tensorflow不再支持。****

由于tensorflow的版本是升级至2.0，已经不再支持本文中需要用到的个别库。并且本文中需要用到的tensorflow1.9同样不支持python3.7。若需要实现本文的所有代码，需要使用对应的软件版本和包版本。给使用者带来了极大的不便