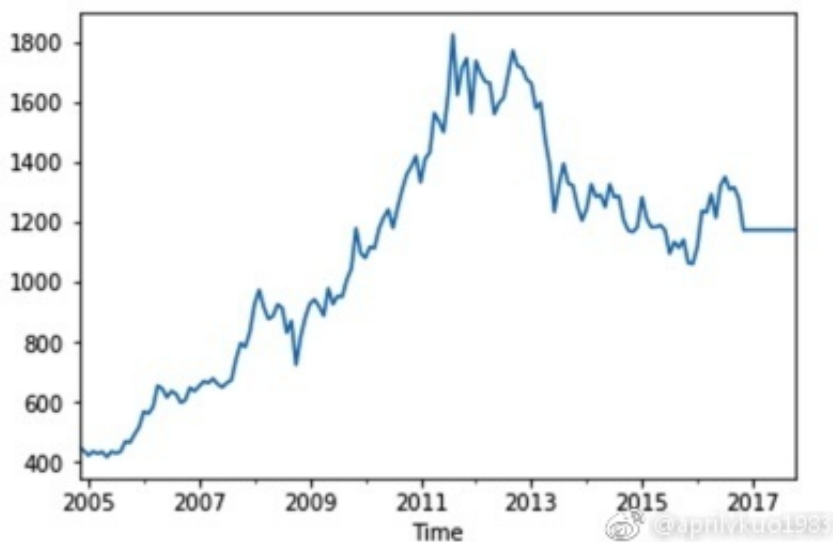


# Rnn on Stocks

训练数据	某金融指标在历史上的价格
用的模型	三层LSTM + Dropout
使用语言框架	python + tensorflow

## 数据预处理

通过pandas绘制的走势图像为：



1. 缺失值用之前的值替代。
2. 用每个月最后一天替代该月的走势

实际观察可以得到，会发现金融指标会在很多天维持在一个数值上面，而且波动幅度较小。可以将时间范围缩小，用每个月的月末作为该月金融指标的走势，可以得到下面的图片。

## 数据规整化：

发现在156个月的原数据值上效果不好，而且收敛较慢，这里采用了缩减策略：每一个月的指标处理这个指标在数据上的平均值，得到156个月的数据。

<b>2017-01-31</b>	1.073020
<b>2017-02-28</b>	1.073020
<b>2017-03-31</b>	1.073020
<b>2017-04-30</b>	1.073020
<b>2017-05-31</b>	1.073020
<b>2017-06-30</b>	1.073020
<b>2017-07-31</b>	1.073020
<b>2017-08-31</b>	1.073020
<b>2017-09-30</b>	1.073020
<b>2017-10-31</b>	1.073020

156 rows x 1 columns



训练数据的产生

## 训练数据的产生

随机生成时间序列，比较时间序列最后一位与它的下一位的变化。如果下一位比他大，那么说明有上升趋势，为正例。反之为负例。

随机产生的时间序列长度为3—30（可调参数）。

## 模型参数

```
learning_rate = 0.001
training_iters = 1000000
batch_size = 128
display_step = 10

# Network Parameters
seq_max_len = 30 # Sequence max length
n_hidden = 100 # hidden layer num of features
n_classes = 2 # linear sequence or not

lstm_cell = tf.contrib.rnn.BasicLSTMCell(n_hidden)
lstm_cell = rnn.DropoutWrapper(cell=lstm_cell, input_keep_prob=1,
output_keep_prob=0.5)
mlstm_cell = rnn.MultiRNNCell([lstm_cell] * 3, state_is_tuple=True)
outputs, states = tf.contrib.rnn.static_rnn(mlstm_cell, x,
dtype=tf.float32,
sequence_length=seqlen)
```

## 运行结果

### 单个lstm

```
Iter 997120, Minibatch Loss= 0.400265, Training Accuracy= 0.78125
Iter 998400, Minibatch Loss= 0.301610, Training Accuracy= 0.84615
Iter 999680, Minibatch Loss= 0.391516, Training Accuracy= 0.75000
Optimization Finished!
Testing Accuracy: 0.634
```

### 三层lstm

```
Iter 997120, Minibatch Loss= 0.183421, Training Accuracy= 0.90625
Iter 998400, Minibatch Loss= 0.338258, Training Accuracy= 0.85577
Iter 999680, Minibatch Loss= 0.245078, Training Accuracy= 0.85938
Optimization Finished!
Testing Accuracy: 0.742
```

## 待改进

1. 没有进一步进行优化调参，交叉验证。
2. 训练数据定义可能不合理
3. 由于是分类问题，损失函数为交叉熵。可以作为回归问题，进行预测下一个月的指标值。损失函数可以设置为困惑度。

4. 模型效果不够理想，毕竟二分类瞎蒙准确率也有50%，缺乏说服力。
5. 评估标准为精度，是否可以进一步用roc，pr等来衡量。
6. 没加入early stopping等。