# 利用LSTM网络和风险估计损失函数做股票交易

## 脚本框架（按照执行顺序）

### 1、rawdata.py

功能：将原始数据转换为按照日期排序的列表。

输入：[stock\_code].csv

输出：sorted\_data （列表类型）

1. 构建RawData类，包含6个变量：date、open、high、close、low、volume
2. 每条记录转换为一个RawData实例。
3. 列表sorted\_data按日期顺序存储实例，并返回。

### 2、chart.py

功能：根据需要提取sorted\_data每一个实例的变量，产生新变量并返回。

输入：sorted\_data

输出：moving\_features，moving\_labels

1. 提取sorted\_data列表的各个实例变量，得到按日期排序的date、open、high、close、low、volume共6个列表（list.shape = [1, data\_len]），并分别转换为numpy.asarray矩阵类型（asarray.shape = [1, data\_len]，其中data\_len等于日期数）。
2. 提供11种类型共61个的特征值，使用6个实例变量矩阵（numpy.asarray）产生新的特征值（利用ta-lib库），并合并为feature（numpy.asarray矩阵类型，shape = [feature\_num, data\_len]）。
3. 设置window值（本例设置为30天），将feature矩阵划分成n个（**x**）[feature\_num, window]矩阵，组成moving\_features（numpy.asarray矩阵，shape = [n, feature\_num, window]）。
4. 如果flatten = True, 利用numpy.ndarray.flatten(“F”)转化为按列展平的一位数组[1, feature\_num \* window]；组成moving\_features（numpy.asarray矩阵，shape = [n, feature\_num \* window]）
5. 同样，设置window值，利用close\_price产生window天内的价格变化值（**y**），得到moving\_labels（numpy.asarray矩阵，shape = [1, n]）。

### 3、smart\_trade.py

功能：构建LSTM(含DropoutWrapper)模型及损失函数。（LSTM结构详细见代码）。

训练模型，最小化LOSS函数（Loss = -100. \* mean(P \* (R - cost))）与优化权值和偏置值，并保存；

利用训练获取的优化参数，预测股票交易。

输入：

num\_step：时间步长数

input\_size：每个时间步长每个对象输入的特征值数量

train\_steps：训练迭代次数

batch\_size：每个时间步长将提供相应的batch\_size个对象

learning\_rate：初始学习速率

hidden\_size：LSTM单元数量

nclasses：输出值的数量

validation\_size：测试集规模

keep\_prob：保留系数（DropoutWrapper）

cost：成本

decay\_step：衰变速度

decay\_rate：衰变系数

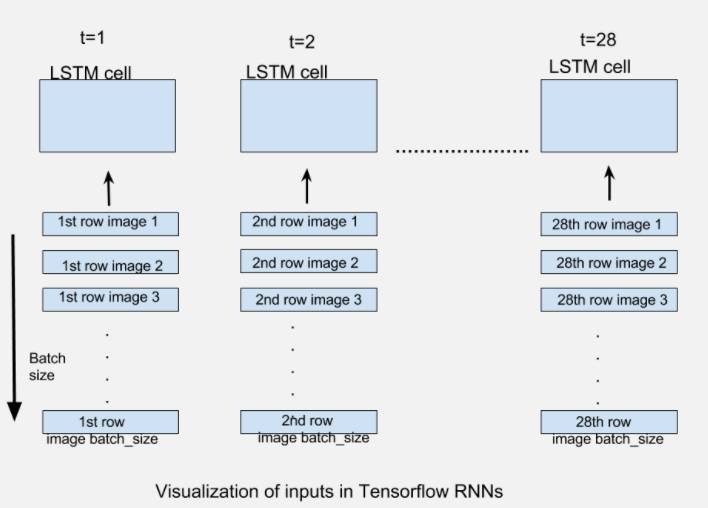
1. 训练：
2. 将特征集合划分成训练集train和测试集val（train + val = n）
3. 将moving\_features分成train\_features（shape = [train, feature\_num, window]）和val\_features（shape = [val, feature\_num, window]）。
4. 将moving\_labels分成train\_labels（shape = [1, train]）和val\_labels（shape = [1, val]）。
5. numpy.transpose()，转置训练集和测试集，得到train\_features（shape = [train, window, feature\_num]）和val\_features（shape = [val, window, feature\_num]）；train\_labels（shape = [train, 1]）和val\_features（shape = [val, 1]）
6. 将train\_features和train\_labels合并在DataSet类实例train\_set，在每次迭代训练中输入batch\_size个x’（转置）和y’（转置）。
7. 设置参数，将train\_set输入LSTM模型进行迭代训练，train\_labels和val\_labels进行优化。
8. 预测：
9. 将moving\_features和moving\_labels转置为shape = [n, window, feature\_num]和shape = [n, 1]。
10. 截取moving\_features和moving\_labels后第一维的val个数据集，合并在DataSet类实例val\_set，在每次迭代训练中输入batch\_size个x’（转置）和y’（转置）。
11. 设置参数，将val\_set输入LSTM模型进行预测。

### 4、dataset.py

功能：对**smart\_trade.py**处理的train\_features和train\_labels整合在DataSet类中，利用next\_batch在每次训练迭代输入batch\_size个x’和y’。batch\_features（shape = [batch\_size, window, feature\_num]）和batch\_lables（shape = [batch\_size, 1]）。

## LSTM模型的输入格式：

对于我们的MNIST图像的情况，我们有大小为28X28的图像。它们可以被推断为具有28行28像素的图像。我们将通过28个时间步骤展开我们的网络，使得在每个时间步长，我们可以输入一行28像素（input\_size），从而通过28个时间步长输入完整的图像。如果我们提供batch\_size图像的数量，每个时间步长将提供相应的batch\_size图像行。下图应该可以解释上述描述：



### 输入形式：

每次迭代输入batch\_features

(shape:[batch\_size, num\_step, input\_size]) 形式，其中

batch\_size: 每个时间步长将提供相应的batch\_size个对象

num\_step: 时间步长数

input\_size: 输入的特征数量

转换成inputs 参数格式: **inputs = tf.unstack(raw\_data ,num\_step,1)**

inputs参数是为了接受形状张量列表[batch\_size,input\_size]。该列表的长度是网络展开的时间步长数num\_step，即该列表的每个元素对应于我们展开网络的相应时间步长的输入。

### 特征值类型：

使用ta-lib库获取特征值。

11大类型共61个特征值：

"ROCP", "OROCP", "HROCP", "LROCP", "MACD", "RSI", "VROCP", "BOLL", "MA", "VMA", "PRICE\_VOLUME"

#### talib.ROCP([price], timeperiod=1)：

变化率百分比（Rate of change Percentage: (price-prevPrice)/prevPrice）

1. ROCP： 收盘价（close）的变化率百分比
2. OROCP：开盘价（open）的变化率百分比
3. HROCP：最高价（high）的变化率百分比
4. LROCP：最低价（low）的变化率百分比
5. VROCP：成交量（volume）的变化率百分比

#### talib.MACD(close, fastperiod=12, slowperiod=26, signalperiod=9)：

MACD称为指数平滑移动平均线，是从双指数移动平均线发展而来，由快的指数移动平均线（EMA12）减去慢的指数移动平均线（EMA26）得到快线DIF，再用2×（快线DIF-DIF的9日加权移动均线DEA）得到MACD柱。

返回：macd 是长短均线的差值（即DIF）；signal是macd的均线（即DEA），如果短均线从下往上突破长均线，为入场信号，进行买入开仓操作；hist是MACD柱。

1. MACD：包含norm\_macd，norm\_signal，norm\_hist，macdrocp，signalrocp，histrocp六个值。

#### talib.RSI(close, timeperiod=n)：

相对强弱指数，Relative Strength Index。

通过比较一段时期内的平均收盘涨数和平均收盘跌数来分析市场买沽盘的意向和实力，从而作出未来市场的走势。

**RSI＝[上升平均数÷(上升平均数＋下跌平均数)]×100**

上升平均数是在某一段日子里升幅数的平均；

下跌平均数则是在同一段日子里跌幅数的平均。

当RSI指标在高位盘整或低位横盘时所出现的各种形态也是判断行情，决定买卖行动的一种分析方法。

1.当RSI曲线在高位(50以上）形成M头或三重顶等高位反转形态时，意味着股价的上升动能已经衰竭，股价有可能出现长期反转行情，投资者应及时地卖出股票。如果股价走势曲线也先后出现同样形态则更可确认，股价下跌的幅度和过程可参照M头或三重顶等顶部反转形态的研判。

2.当RSI曲线在低位(50以下）形成W底或三重底等低位反转形态时，意味着股价的下跌动能已经减弱，股价有可能构筑中长期底部，投资者可逢低分批建仓。如果股价走势曲线也先后出现同样形态则更可确认，股价的上涨幅度及过程可参照W底或三重底等底部反转形态的研判。

3.RSI曲线顶部反转形态对行情判断的准确性要高于底部形态

1. RSI：包括rsi6，rsi12，rsi24，rsi6rocp，rsi12rocp，rsi24rocp
2. BOLL：

布林线（Boll）指标是股市技术分析的常用工具之一，通过计算股价的“标准差”，再求股价的“信赖区间”。

该指标在图形上画出三条线，其中上下两条线可以分别看成是股价的压力线和支撑线，而在两条线之间还有一条股价平均线，布林线指标的参数最好设为20。一般来说，股价会运行在压力线和支撑线所形成的通道中。

日BOLL指标的计算公式

* 中轨线（MD）=N日的移动平均线
* 上轨线（UP）=中轨线+两倍的标准差
* 下轨线（DN）=中轨线－两倍的标准差

#### talib.MA([close, volume], timeperiod=n)：

滑动平均（moving average），是做时间序列预测时用到的简单方法。

计算方法：对于一个给定的数列，首先设定一个固定的值k，然后分别计算第1项到第k项，第2项到第k+1项，第3项到第k+2项的平均值，依次类推。

1. MA：收盘价的滑动平均，包括ma5，ma10，ma20，ma30，ma60，ma90，ma120，ma180，ma360，ma720；ma5rocp，ma10rocp，ma20rocp，ma30rocp，ma60rocp，ma90rocp，ma120rocp，ma180rocp，ma360rocp，ma720rocp。
2. VMA：成交量的滑动平均，包括ma5，ma10，ma20，ma30，ma60，ma90，ma120，ma180，ma360，ma720；ma5rocp，ma10rocp，ma20rocp，ma30rocp，ma60rocp，ma90rocp，ma120rocp，ma180rocp，ma360rocp，ma720rocp。

#### pv = rocp \* vrocp:

1. PRICE\_VOLUME：包含pv