

LSTM 深度学习模型进行日内股指预测报告 V2.0

(对应代码: LSTM_V4.0)

一、对 V1.0 的改进

针对“预测报告 V1.0”中提到的不足之处，V2.0 中进行了如下修改：

1.对迭代过多造成程序中断错误的改进

“预测报告 V1.0”中，当迭代次数过多时，会出现“每次传入的数组大小不一致”导致程序中断的问题。经过不断调试，找到了问题所在：在 next_batch 中，每次都是在上一步的 n 条数据的基础上，取后面的 n 条数据，取完之后没有返回到列表第一条数据，导致后面的每一个 batch 都为空值。

该版本中，使用随机取值的方法，每一个 batch，都随机取数据集中的某一段。同时，减小每个 batch 的长度，使得取值的随机性更大。这样保证程序可以无限制增加迭代次数，并且使数据集分的更精细。

2.对连续分钟预测的改进

V1.0 中，只能根据当前时间预测下一分钟的涨跌情况，还不能根据前面几分钟的数据预测后面几分钟的涨跌情况。这对于策略编写来说还远远不足。

该版本中，在数据集中增加了当前时刻后 1 分钟涨跌幅、2 分钟涨跌幅……5 分钟涨跌幅等五个特征，并根据这五个特征分别建立标签进行分类，然后分别进行训练和测试，以求得测试准确率。

同时，我们是根据这五个指标进行打标签，所以在数据集中应把这五个指标去掉，而是让神经网络模型主动去训练寻找特征，防止这些指标对预测分类带来太大的主导作用。

3.对涨跌幅持续时间预测的改进

基于当前分钟对下一分钟的预测，统计涨跌幅持续时间。根据预测值，对每一个大涨趋势和大跌趋势计算了持续时间，并且运用了一个比较宽松的公式计算了预测准确率。

但是，对涨跌幅持续时间预测并不是一个写策略的好选择。举例来说，将持续时间分成 n 类，其中有一类为“10-12 分钟”。如果 10 分钟之前都是上涨趋势，而在 11 分钟后出现了大跌趋势，在这个宽泛的范围内就很难作出正确决策。但如果分的太细，具体到分钟，会使得分类的准确率很低。因此这个预测暂时搁置。

4.对添加 RSI、布林带指标等的改进(对应代码: LSTM_feature_adding)

(1) RSI: 相对强弱指数 RSI 是根据一定时期内上涨点数和涨跌点数之和的比率制作出的一种技术曲线。能够反映出市场在一定时期内的景气程度，非常适合做短线差价操作。

强弱指标理论认为，任何市价的大涨或大跌，均在 0-100 之间变动，根据常态分配，认为 RSI 值多在 30-70 之间变动，通常 80 甚至 90 时被认为市场已到达超买状态，至此市场价格自然会回落调整。当价格低跌至 30 以下即被认为是超卖状态，市价将出现反弹回升。

由于时间关系，RSI 指标暂未添加，也没有对该指标进行预测。待策略编写完成后，再逐步对模型进行优化。

(2) 布林带：它由三条轨道线组成，其中上下两条线分别可以看成是价格的压力线和支撑线，在两条线之间是一条价格平均线，一般情况价格线在由上下轨道组成的带状区间游走，而且随价格的变化而自动调整轨道的位置。当波带变窄时，激烈的价格波动有可能随即产生；若高低点穿越带边线时，立刻又回到波带内，则会有回档产生。

三条线分别为：中轨线：N 个单位的移动平均线；上轨线：中轨线 + 两倍的标准差；下轨线：中轨线 - 两倍的标准差。

添加完布林带后，数据集加入了四个特征：上轨值、下轨值、%b 指标、通道宽度。

5.对预测移动平均线、布林带指标的改进

除了预测连续分钟的涨跌情况，还可以预测移动平均线、布林带等指标，以便写出更好的策略。

二、改进后的结果

1.连续分钟预测的准确率

(1) 当前时刻后 1 分钟，73.5%

```
Iter 84000, Minibatch Loss= 0.815975, Training Accuracy= 0.70000
Iter 86000, Minibatch Loss= 0.404420, Training Accuracy= 0.85000
Iter 88000, Minibatch Loss= 0.868310, Training Accuracy= 0.60000
Iter 90000, Minibatch Loss= 0.850256, Training Accuracy= 0.70000
Iter 92000, Minibatch Loss= 0.866675, Training Accuracy= 0.60000
Iter 94000, Minibatch Loss= 1.250295, Training Accuracy= 0.45000
Iter 96000, Minibatch Loss= 0.866472, Training Accuracy= 0.60000
Iter 98000, Minibatch Loss= 1.051986, Training Accuracy= 0.50000
Optimization Finished!
('Testing Accuracy:', 0.73500001)
```

(2) 当前时刻后 2 分钟，68.8%

```
Iter 82000, Minibatch Loss= 0.769406, Training Accuracy= 0.70000
Iter 84000, Minibatch Loss= 0.833936, Training Accuracy= 0.65000
Iter 86000, Minibatch Loss= 0.704227, Training Accuracy= 0.85000
Iter 88000, Minibatch Loss= 0.793097, Training Accuracy= 0.65000
Iter 90000, Minibatch Loss= 0.882792, Training Accuracy= 0.70000
Iter 92000, Minibatch Loss= 1.028928, Training Accuracy= 0.50000
Iter 94000, Minibatch Loss= 0.885114, Training Accuracy= 0.60000
Iter 96000, Minibatch Loss= 1.004381, Training Accuracy= 0.70000
Iter 98000, Minibatch Loss= 0.931924, Training Accuracy= 0.55000
Optimization Finished!
('Testing Accuracy:', 0.68800002)
```

(3) 当前时刻后 3 分钟，68.3%

```
Iter 82000, Minibatch Loss= 0.700553, Training Accuracy= 0.65000
Iter 84000, Minibatch Loss= 0.652189, Training Accuracy= 0.70000
Iter 86000, Minibatch Loss= 1.221922, Training Accuracy= 0.35000
Iter 88000, Minibatch Loss= 0.983770, Training Accuracy= 0.40000
Iter 90000, Minibatch Loss= 0.791664, Training Accuracy= 0.70000
Iter 92000, Minibatch Loss= 0.981209, Training Accuracy= 0.50000
Iter 94000, Minibatch Loss= 0.880174, Training Accuracy= 0.65000
Iter 96000, Minibatch Loss= 0.925318, Training Accuracy= 0.60000
Iter 98000, Minibatch Loss= 0.765606, Training Accuracy= 0.75000
Optimization Finished!
('Testing Accuracy:', 0.68300003)
```

(4) 当前时刻后 4 分钟, 59.85%

```
Iter 82000, Minibatch Loss= 0.923656, Training Accuracy= 0.70000
Iter 84000, Minibatch Loss= 0.960349, Training Accuracy= 0.50000
Iter 86000, Minibatch Loss= 0.918003, Training Accuracy= 0.55000
Iter 88000, Minibatch Loss= 0.940913, Training Accuracy= 0.65000
Iter 90000, Minibatch Loss= 0.984426, Training Accuracy= 0.45000
Iter 92000, Minibatch Loss= 0.976268, Training Accuracy= 0.40000
Iter 94000, Minibatch Loss= 0.736631, Training Accuracy= 0.60000
Iter 96000, Minibatch Loss= 0.995859, Training Accuracy= 0.60000
Iter 98000, Minibatch Loss= 0.780405, Training Accuracy= 0.85000
Optimization Finished!
('Testing Accuracy:', 0.59850001)
```

(5) 当前时刻后 5 分钟, 44.75%

```
Iter 82000, Minibatch Loss= 1.087931, Training Accuracy= 0.65000
Iter 84000, Minibatch Loss= 1.182249, Training Accuracy= 0.55000
Iter 86000, Minibatch Loss= 1.025599, Training Accuracy= 0.75000
Iter 88000, Minibatch Loss= 1.412207, Training Accuracy= 0.30000
Iter 90000, Minibatch Loss= 1.222993, Training Accuracy= 0.45000
Iter 92000, Minibatch Loss= 1.313014, Training Accuracy= 0.50000
Iter 94000, Minibatch Loss= 0.918286, Training Accuracy= 0.60000
Iter 96000, Minibatch Loss= 1.028222, Training Accuracy= 0.65000
Iter 98000, Minibatch Loss= 1.272557, Training Accuracy= 0.25000
Optimization Finished!
('Testing Accuracy:', 0.44749999)
```

2.连续分钟预测的分析

- (1) 对模型进行一定的优化之后, 预测下一分钟的准确率有了一定的提升。
- (2) 随着预测时间的延长, 准确率会依次降低, 符合实际情况。
- (3) 预测后五分钟的准确率仍能达到 45%左右, 说明该模型能起到不错的预测效果。

3.涨跌趋势持续时间的准确率

(1) 预测依据: 一般的交易选择应当是: 若预测到上涨趋势, 在开始上涨时建多仓, 在即将下跌时平多仓; 若预测到下跌趋势, 在开始下跌时建空仓, 在即将上涨时平空仓。

同时, 大涨趋势和大跌趋势是比较重要的两种趋势, 在交易选择时扮演着相当重要的角色。

因此我选择根据预测值, 从下面四个方面来进行趋势持续时间的准确率验证。

- 1) 出现大涨趋势时, 涨持时间 (下一个出现下一个下跌趋势之前的时间)
- 2) 出现小涨趋势时, 涨持时间 (下一个出现下一个下跌趋势之前的时间)
- 3) 出现小跌趋势时, 跌持时间 (下一个出现下一个下跌趋势之前的时间)
- 4) 出现大跌趋势时, 跌持时间 (下一个出现下一个下跌趋势之前的时间)

(2) 预测结果

1) 大涨:

预测大涨持续时间:

第 60 分钟为大涨趋势, 直到 6 分钟后开始下跌
第 159 分钟为大涨趋势, 直到 2 分钟后开始下跌
第 165 分钟为大涨趋势, 直到 3 分钟后开始下跌
第 166 分钟为大涨趋势, 直到 2 分钟后开始下跌
第 184 分钟为大涨趋势, 直到 3 分钟后开始下跌
第 235 分钟为大涨趋势, 直到 17 分钟后开始下跌
第 247 分钟为大涨趋势, 直到 5 分钟后开始下跌
第 248 分钟为大涨趋势, 直到 4 分钟后开始下跌
第 249 分钟为大涨趋势, 直到 3 分钟后开始下跌
第 250 分钟为大涨趋势, 直到 2 分钟后开始下跌
第 256 分钟为大涨趋势, 直到 18 分钟后开始下跌
第 385 分钟为大涨趋势, 直到 8 分钟后开始下跌
第 386 分钟为大涨趋势, 直到 7 分钟后开始下跌
第 390 分钟为大涨趋势, 直到 3 分钟后开始下跌
第 427 分钟为大涨趋势, 直到 3 分钟后开始下跌
第 428 分钟为大涨趋势, 直到 2 分钟后开始下跌
第 536 分钟为大涨趋势, 直到 3 分钟后开始下跌
第 810 分钟为大涨趋势, 直到 3 分钟后开始下跌

2) 大涨预测准确率:

accuary: 0.318181818182

3) 大跌:

预测大跌持续时间:

第 19 分钟为大跌趋势, 直到 3 分钟后开始上涨
第 121 分钟为大跌趋势, 直到 1 分钟后开始上涨
第 126 分钟为大跌趋势, 直到 4 分钟后开始上涨
第 131 分钟为大跌趋势, 直到 2 分钟后开始上涨
第 187 分钟为大跌趋势, 直到 1 分钟后开始上涨
第 632 分钟为大跌趋势, 直到 3 分钟后开始上涨
第 673 分钟为大跌趋势, 直到 1 分钟后开始上涨
第 783 分钟为大跌趋势, 直到 2 分钟后开始上涨
第 957 分钟为大跌趋势, 直到 3 分钟后开始上涨
第 1188 分钟为大跌趋势, 直到 2 分钟后开始上涨
第 1377 分钟为大跌趋势, 直到 1 分钟后开始上涨
第 1466 分钟为大跌趋势, 直到 2 分钟后开始上涨
第 1526 分钟为大跌趋势, 直到 2 分钟后开始上涨
第 1546 分钟为大跌趋势, 直到 2 分钟后开始上涨
第 1699 分钟为大跌趋势, 直到 1 分钟后开始上涨
第 1913 分钟为大跌趋势, 直到 3 分钟后开始上涨
第 1914 分钟为大跌趋势, 直到 2 分钟后开始上涨
第 1915 分钟为大跌趋势, 直到 1 分钟后开始上涨

4) 大跌预测准确率:

accuary: 0.833333333333

4.涨跌趋势持续时间预测的分析

- (1) 发现大涨趋势的持续时间比大跌趋势的持续时间长，准确率也较后者偏低。
- (2) 持续时间的计算，只是基于当前分钟对下一分钟的预测，统计涨跌幅持续时间而来。在历史数据中方便运用，但用在正式的实时交易中，比较困难。并且如上文中提到，分类过细和过于宽泛都对策略编写造成较大影响，因此暂时不考虑使用该方面的预测。

5.下一分钟移动平均线预测准确率

统计了MA_5、MA_12、MA_26、布林带上轨线、下轨线在数据集中的最大值和最小值，以布林带下轨线为例。

```
# 统计最大值和最小值，便于划分区间
data3 = data[1000:11000]
max(data3['boll_down_PRED']),min(data3['boll_down_PRED'])

(3230.1575954099999, 2739.6132079899999)
```

发现五个指标的值均在一个固定的区间内。因此对数据值进行分类打标签，共分为 10 类。这十类分别为：<2800，2800~2850,2850~2900，……，3150~3200，>3200。同样带入我们的模型中进行预测并输出结果。得到准确率分别如下所示：

- (1) MA_5, 60.95%

```
Iter 82000, Minibatch Loss= 0.031548, Training Accuracy= 1.00000
Iter 84000, Minibatch Loss= 0.026963, Training Accuracy= 1.00000
Iter 86000, Minibatch Loss= 0.460857, Training Accuracy= 0.75000
Iter 88000, Minibatch Loss= 1.049788, Training Accuracy= 0.00000
Iter 90000, Minibatch Loss= 0.026651, Training Accuracy= 1.00000
Iter 92000, Minibatch Loss= 0.052637, Training Accuracy= 1.00000
Iter 94000, Minibatch Loss= 0.355596, Training Accuracy= 0.85000
Iter 96000, Minibatch Loss= 1.213719, Training Accuracy= 0.00000
Iter 98000, Minibatch Loss= 0.602614, Training Accuracy= 0.70000
Optimization Finished!
('Testing Accuracy:', 0.60949999)
```

- (2) MA_12, 56.6%

```
Iter 82000, Minibatch Loss= 0.261556, Training Accuracy= 1.00000
Iter 84000, Minibatch Loss= 0.186185, Training Accuracy= 1.00000
Iter 86000, Minibatch Loss= 0.127413, Training Accuracy= 1.00000
Iter 88000, Minibatch Loss= 0.421723, Training Accuracy= 0.90000
Iter 90000, Minibatch Loss= 0.439835, Training Accuracy= 0.75000
Iter 92000, Minibatch Loss= 0.070634, Training Accuracy= 1.00000
Iter 94000, Minibatch Loss= 0.118000, Training Accuracy= 1.00000
Iter 96000, Minibatch Loss= 0.553685, Training Accuracy= 0.70000
Iter 98000, Minibatch Loss= 0.294946, Training Accuracy= 1.00000
Optimization Finished!
('Testing Accuracy:', 0.56599998)
```

(3) MA_26, 56.1%

```
Iter 82000, Minibatch Loss= 0.186962, Training Accuracy= 1.00000
Iter 84000, Minibatch Loss= 0.917571, Training Accuracy= 0.15000
Iter 86000, Minibatch Loss= 0.049315, Training Accuracy= 1.00000
Iter 88000, Minibatch Loss= 0.115120, Training Accuracy= 1.00000
Iter 90000, Minibatch Loss= 0.015581, Training Accuracy= 1.00000
Iter 92000, Minibatch Loss= 0.037905, Training Accuracy= 1.00000
Iter 94000, Minibatch Loss= 1.164464, Training Accuracy= 0.25000
Iter 96000, Minibatch Loss= 0.427588, Training Accuracy= 0.75000
Iter 98000, Minibatch Loss= 0.007139, Training Accuracy= 1.00000
Optimization Finished!
('Testing Accuracy:', 0.56099999)
```

6.下一分钟布林带预测的准确率

(1) 上轨线, 68.2%

```
Iter 82000, Minibatch Loss= 0.091667, Training Accuracy= 1.00000
Iter 84000, Minibatch Loss= 0.422920, Training Accuracy= 1.00000
Iter 86000, Minibatch Loss= 0.266617, Training Accuracy= 1.00000
Iter 88000, Minibatch Loss= 0.205343, Training Accuracy= 1.00000
Iter 90000, Minibatch Loss= 0.045159, Training Accuracy= 1.00000
Iter 92000, Minibatch Loss= 0.107823, Training Accuracy= 1.00000
Iter 94000, Minibatch Loss= 0.579919, Training Accuracy= 0.80000
Iter 96000, Minibatch Loss= 0.015893, Training Accuracy= 1.00000
Iter 98000, Minibatch Loss= 0.375310, Training Accuracy= 0.90000
Optimization Finished!
('Testing Accuracy:', 0.68199998)
```

(2) 下轨线, 47.9%

```
Iter 82000, Minibatch Loss= 0.560647, Training Accuracy= 0.80000
Iter 84000, Minibatch Loss= 0.585663, Training Accuracy= 0.75000
Iter 86000, Minibatch Loss= 0.558475, Training Accuracy= 0.65000
Iter 88000, Minibatch Loss= 0.124142, Training Accuracy= 1.00000
Iter 90000, Minibatch Loss= 0.554778, Training Accuracy= 0.60000
Iter 92000, Minibatch Loss= 0.319227, Training Accuracy= 0.90000
Iter 94000, Minibatch Loss= 0.523589, Training Accuracy= 0.65000
Iter 96000, Minibatch Loss= 0.652594, Training Accuracy= 0.70000
Iter 98000, Minibatch Loss= 0.130993, Training Accuracy= 0.95000
Optimization Finished!
('Testing Accuracy:', 0.479)
```

三、下一步目标

至此, 利用 LSTM 对未来数据的预测已经基本成型。下一步先结合模型进行策略编写, 看看效果如何。然后针对不足之处再回头进行修改和优化。

(Powered by Feng Zipeng)