

国债期货量化报告——几种策略的实现和回测

国债期货量化报告——几种策略的实现和回测

H1

数据参数和数据处理

数据描述

数据处理

简单神经网络

模型介绍

模型参数

模型结果

LSTM

模型介绍

模型参数

模型结果

xgboost

模型介绍

模型参数

模型结果

使用softmax转变为分类问题

简单神经网络

LSTM

后期优化方向

本文我们使用几种不同的量化模型进行国债期货的量化

小结：

总体来说，在模型的测试集内，机器学习模型能够有一定的超额收益率，机器学习模型预测结果在累计收益率、胜率、最大回撤等方面表现优于国债期货指数。

但是限于机器学习的局限性，短期内机器学习模型的表现并不一定优于国债期货指数，且当前的回测结果并不能保证未来模型的表现。

数据参数和数据处理

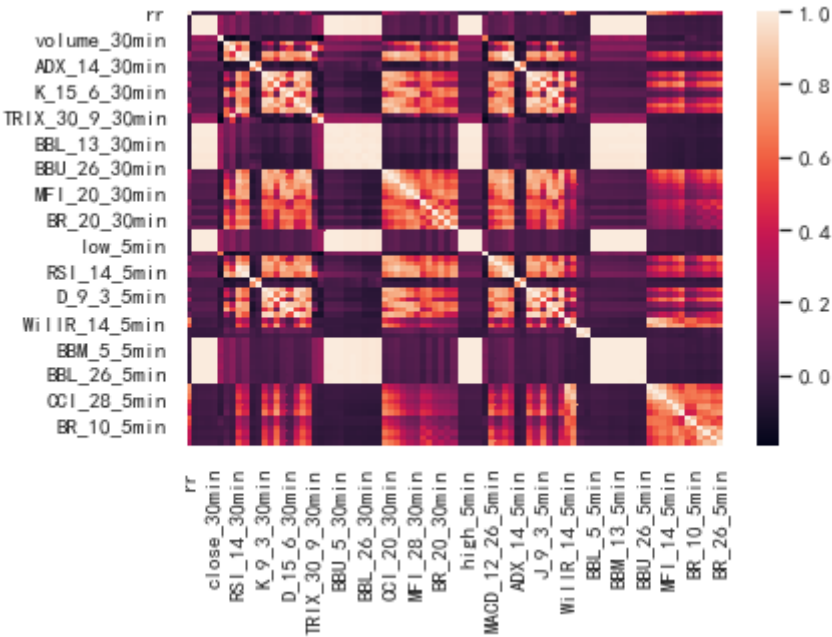
H3 数据描述

H2

我们使用国债期货的30min数据，时间范围为从2018年末至今，使用的数据包括如下的参数，同时每一个参数分为30min和5min两个维度。

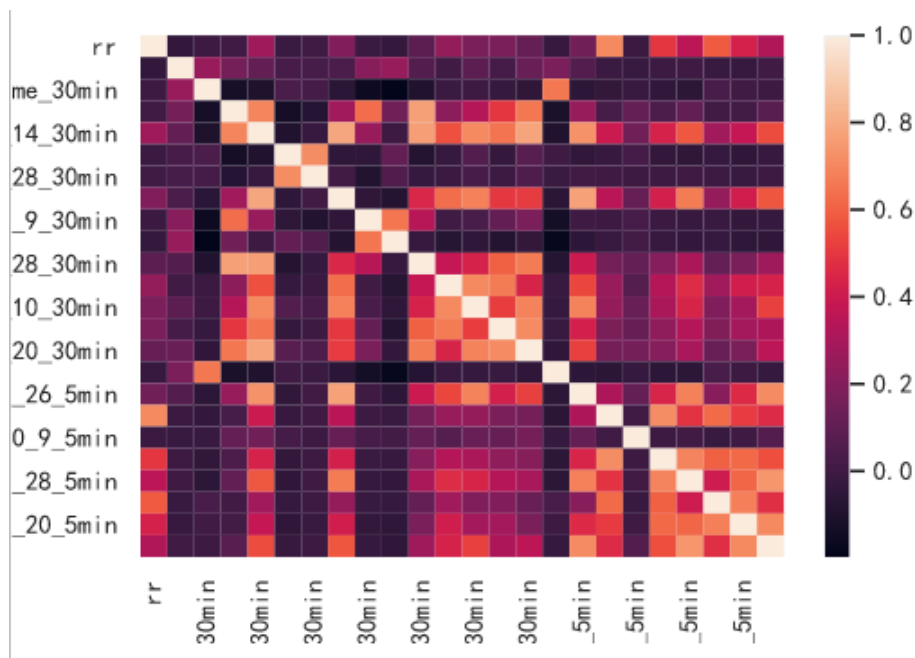
open_30min	RSI_14_30min	TRIX_30_9_30min	CCI_14_30min
high_30min	RSI_28_30min	TRIX_60_15_30min	CCI_20_30min
low_30min	ADX_14_30min	BBL_5_30min	CCI_28_30min
close_30min	ADX_28_30min	BBM_5_30min	MFI_14_30min
volume_30min	K_9_3_30min	BBU_5_30min	MFI_20_30min
WillR_14_30min	D_9_3_30min	BBL_13_30min	MFI_28_30min
WillR_30_30min	J_9_3_30min	BBM_13_30min	AR_10_30min
MACD_12_26_30min	K_15_6_30min	BBU_13_30min	BR_10_30min
MACD_24_52_30min	D_15_6_30min	BBL_26_30min	AR_20_30min
	J_15_6_30min	BBM_26_30min	BR_20_30min
		BBU_26_30min	AR_26_30min
			BR_26_30min

参数共有84个，其相关系数矩阵如下所示



可以看到许多参数之间有过高的相关系数，我们采用简单的降维方式，对于相关系数超过0.8的高相关性参数，我们仅保留一个。

经过处理后，相关系数矩阵如下图所示，可以看到不存在过高的相关系数



H3 数据处理

对于时间序列数据，将其转变为自变量、因变量的集合，对于T时期的数据，我们使用过去10个时期的数据对其进行预测，因此对于一个数据，输入的数据总维度为 $24 \times 10 = 240$ 维

```

1  def create_dataset(dataset:np.ndarray, look_back:int): # 后一个数据和前
    look_back个数据有关系
2      assert not (isinstance(dataset,np.ndarray) and
    isinstance(dataset,pd.core.frame.DataFrame)) , '输入数据格式不符合要求，使用
    pandas或numpy'
3      dataset=np.array(dataset)
4
5      dataX, dataY = [],[]
6      for i in range(len(dataset) - look_back - 1):
7          a = dataset[i:(i + look_back)]
8          dataX.append(a) #
9          dataY.append(dataset[:,0][i + look_back]) #此处Y要第一列（收益率）
10
11     dataX,dataY=np.array(dataX),np.array(dataY)
12     dataX,dataY=np.reshape(dataX,(dataX.shape[0],-1)),np.reshape(dataY,
    (-1,1))# 转变为二维数据
13
14     # 进行数据归一化
15     from sklearn.preprocessing import MinMaxScaler
16
17     scalerX = MinMaxScaler(feature_range=(0, 1)) # 归一化X
18     scalerX.fit(dataX)

```

```

19     dataX=scalerX.transform(dataX)
20
21     scalerY = MinMaxScaler(feature_range=(0, 1)) # 归一化Y
22     scalerY.fit(dataY)
23     dataY=scalerY.transform(dataY)
24
25     return np.array(dataX), np.array(dataY), scalerX, scalerY # 生成输入数据
    和输出数据

```

将数据划分为训练集和测试集，其中训练集为前2/3的数据，测试集为后1/3的数据

```

1  def split_dataset(dataset):
2      dataset=np.array(dataset)
3      train_size = int(len(dataset) * 0.67) # %67的训练集，剩下测试集
4      test_size = len(dataset) - train_size
5      train, test = dataset[0:train_size, :],
        dataset[train_size:len(dataset), :] # 训练集和测试集
6      return train,test
7
8  trainX,testX=split_dataset(totX)
9  trainY,testY=split_dataset(totY)

```

简单神经网络

H3 模型介绍

H2

ANN是指由大量的处理单元(神经元) 互相连接而形成的复杂网络结构，是对人脑组织结构和运行机制的某种抽象、简化和模拟。 人工神经网络（Artificial Neural Network，简称ANN），以数学模型模拟神经元活动，是基于模仿大脑神经网络结构和功能而建立的一种信息处理系统。

H3 模型参数

```

1  tBatchSize = 128
2  '''第一步：选择模型'''
3  model = Sequential() # 采用贯序模型
4
5  '''第二步：构建网络层'''
6  '''构建网络只是构建了一个网络结构，并定义网络的参数，此时还没有输入的数据集'''

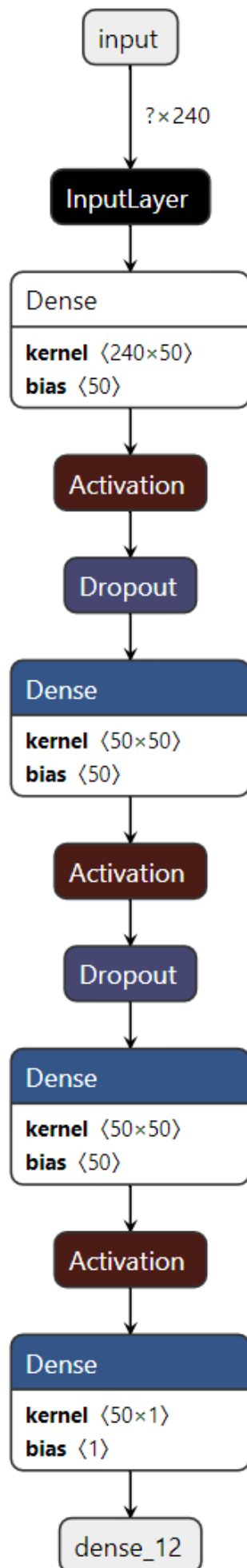
```

```

7  # 构建的第一个层作为输入层
8  model.add(Dense(50, input_shape=(trainX.shape[1],)))
9  model.add(Activation('relu')) # 激活函数是relu
10 model.add(Dropout(0.5)) # 采用50%的dropout 随机取一半进行训练
11
12 # 构建的第2个层作为隐藏层2，（如果加上输入层，实际上是第三层）
13 model.add(Dense(50)) # 隐藏层节点50个
14 model.add(Activation('relu'))
15 model.add(Dropout(0.5))
16
17 model.add(Dense(50)) # 隐藏层3，节点50个
18 model.add(Activation('relu'))
19
20 # 构建的第3个层作为输出层
21 model.add(Dense(1)) #
22
23 '''第三步：网络优化和编译'''
24
25 sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True) # 优化函数，设
    定学习率（lr）等参数
26 model.compile(loss='mean_squared_error', optimizer=sgd) # 均方误差作为损失函
    数，优化器为sgd

```

模型的可视化：



H3 模型结果

我们使用训练集中训练得到的模型对测试集数据进行预测，得到T时刻的收益率的预测值，我们将策略结果分为两个

- 1. 如果收益率预测值>0，我们考虑买入并持有期货，预测收益率<0不做操作
- 2. 如果收益率预测值>0，我们考虑买入并持有期货，预测收益率<0卖空期货

从下图结果我们可以看到两个策略均能在回测区间内累计收益率超过估值期货，其中同时进行多仓和空仓交易能够产生更好的累计收益率，说明模型对国债期货的上行和下行均有一定的预测准确度。

但同时我们也可以看到在短期内，如最初的50个时间期间，模型的表现仍有可能弱于期货收益率。



	期货	策略（多仓）	策略（多空仓）
信息比率	-0.1573	0.6044	0.7311
累计收益率	0.9892	1.0193	1.0496
每日收益率	0.0000	0.0000	0.0000
最大每日收益	0.39%	0.26%	0.34%
最大每日损失	-0.34%	-0.26%	-0.39%
上涨天数	689	148	751
下跌天数	726	123	664
不变天数	51	1195	51

	期货	策略（多仓）	策略（多空仓）
胜率	48.69%	54.61%	53.07%
上涨时平均每日收益率	0.00057	0.00062	0.00056
下跌时平均每日收益率	-0.00055	-0.00059	-0.00056
盈亏比	1.026089	1.052196	1.000341
最大回撤	3.69%	0.84%	1.47%

LSTM

H3 模型介绍

H2

长短期记忆网络（[LSTM](#)，Long Short-Term Memory）是一种时间循环神经网络，是为了解决一般的[RNN](#)（[循环神经网络](#)）存在的长期依赖问题而专门设计出来的，所有的RNN都具有一种重复神经网络模块的链式形式。在标准RNN中，这个重复的结构模块只有一个非常简单的结构，例如一个tanh层。

H3 模型参数

```
1 model = Sequential()
2 model.add(LSTM(11, input_shape=(24,10))) # 隐层11个神经元，输入数据格式为24维*10天
3 model.add(Dense(1))
4 model.compile(loss='mse', optimizer='adam') # 评价函数mse，优化器adam
5 model.fit(trainX_LSTM, trainY, epochs=50, batch_size=100, verbose=1) # 100次迭代
```

H3 模型结果



xgboost

H3 模型介绍

H2

XGBoost是"极端梯度提升"(eXtreme Gradient Boosting)的简称。XGBoost 源于梯度提升框架，但是更加高效，秘诀就在于算法能并行计算、近似建树、对稀疏数据的有效处理以及内存使用优化，这使得 XGBoost 至少比现有梯度提升实现有至少 10 倍的速度提升。

H3 模型参数

```
1  # 参数设置
2  params={
3      'booster':'gbtree',
4      'objective':'reg:squarederror',
5      'eval_metric': 'auc',
6      'max_depth':4,
7      'lambda':10,
8      'subsample':0.75,
9      'colsample_bytree':0.75,
10     'min_child_weight':2,
11     'eta': 0.025,
12     'seed':0, #随机数的种子
13     # 'nthread':8, #XGBoost运行时的线程数。缺省值是当前系统可以获得的最大线程数
14 }
```

H3 模型结果



	期货	策略（多仓）	策略（多空仓）
信息比率	-0.1573	0.3641	0.7309
累计收益率	0.9892	1.0191	1.0496
每日收益率	0.0000	0.0000	0.0000
最大每日收益	0.0039	0.0026	0.0027
最大每日损失	-0.0034	-0.0034	-0.0039
上涨天数	689	436	725
下跌天数	726	437	690
不变天数	51	593	51
胜率	0.48693	0.49943	0.51237
上涨时平均每日收益率	0.00057	0.00058	0.00058
下跌时平均每日收益率	-0.00055	-0.00054	-0.00054

使用softmax转变为分类问题

H2 由于我们在上述的过程中，主要只是使用收益率的正负值，因此我们通过softmax函数将其转变为分类问题，在输入参数中，如果当日收益率为正，则记为1，若收益率为负，记为0，模型可以输出待预测时期的0/1的概率，我们选择概率较大的类别，以此作为预测值

H3 简单神经网络



	期货	策略（多仓）	策略（多空仓）
信息比率	-0.1473	0.1192	0.3362
累计收益率	0.9898	1.0061	1.0223
每日收益率	0.0000	0.0000	0.0000
最大每日收益	0.0039	0.0039	0.0039
最大每日损失	-0.0034	-0.0026	-0.0026
上涨天数	688	455	707
下跌天数	724	472	705
不变天数	51	536	51
胜率	0.48725	0.49083	0.50071
上涨时平均每日收益率	0.00057	0.00057	0.00058
下跌时平均每日收益率	-0.00055	-0.00053	-0.00055
盈亏比	1.026472	1.063379	1.055442
最大回撤	0.0369	0.0143	0.0166

H3 LSTM



	期货	策略（多仓）	策略（多空仓）
信息比率	-0.1473	0.2848	0.5570
累计收益率	0.9898	1.0136	1.0374
每日收益率	0.0000	0.0000	0.0000
最大每日收益	0.0039	0.0024	0.0029
最大每日损失	-0.0034	-0.0034	-0.0039
上涨天数	688	410	739
下跌天数	724	395	673
不变天数	51	658	51
胜率	0.48725	0.50932	0.52337
上涨时平均每日收益率	0.00057	0.00055	0.00056
下跌时平均每日收益率	-0.00055	-0.00053	-0.00056
盈亏比	1.026472	1.026056	1.000529
最大回撤	0.0369	0.0206	0.0168

后期优化方向

H2

- 损失函数考虑调整为更能反映策略收益率的指标
- 输入数据的特征优化
- 模型的神经网络参数优化等
- 基于模型使用现有数据进行模拟运行