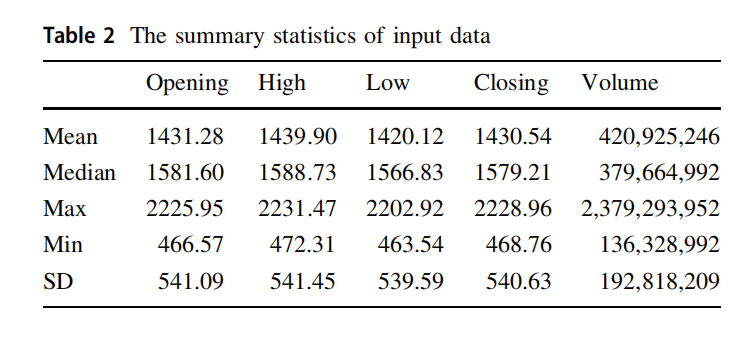
## **数据处理**

##### **数据集准备**

选取2000年1月4日至2016年12月31日的daily KOSPI数据（共计17年），数据源自Yahoo Finance。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Date | Open | High | Low | Close | Adj Close | Volume |
| 2022/10/11 | 2193.02002 | 2193.050049 | 2174.060059 | 2192.070068 | 2192.070068 | 859100 |
| 2022/10/12 | 2191.350098 | 2205.02002 | 2181.649902 | 2202.469971 | 2202.469971 | 658000 |
| 2022/10/13 | 2198.290039 | 2198.340088 | 2162.870117 | 2162.870117 | 2162.870117 | 792200 |
| 2022/10/14 | 2200.659912 | 2220.540039 | 2193.629883 | 2212.550049 | 2212.550049 | 512900 |

论文原数据集分布如下：



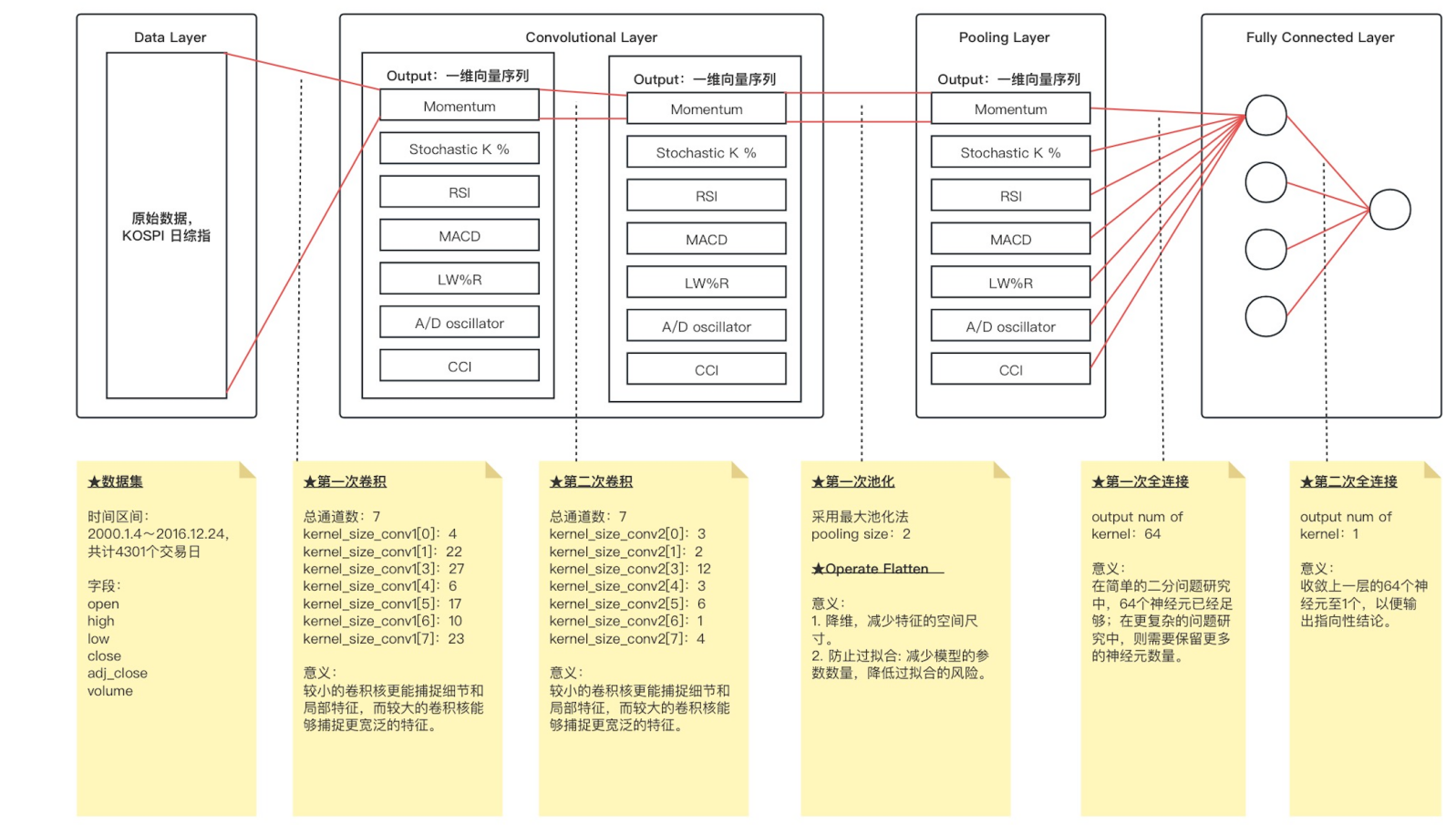
本文所采用数据集分布如下：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Open | High | Low | Close | Adj Close | Volume |
| Mean | 1431.22 | 1439.84 | 1420.06 | 1430.49 | 1430.49 | 421388.03 |
| Median | 1581.60 | 1588.73 | 1566.83 | 1579.21 | 1579.21 | 379800.00 |
| Max | 2225.95 | 2231.47 | 2202.92 | 2228.96 | 2228.96 | 2379300.00 |
| Min | 466.57 | 472.31 | 463.54 | 468.76 | 468.76 | 136300.00 |
| SD | 541.27 | 541.62 | 539.77 | 540.80 | 540.80 | 192935.96 |

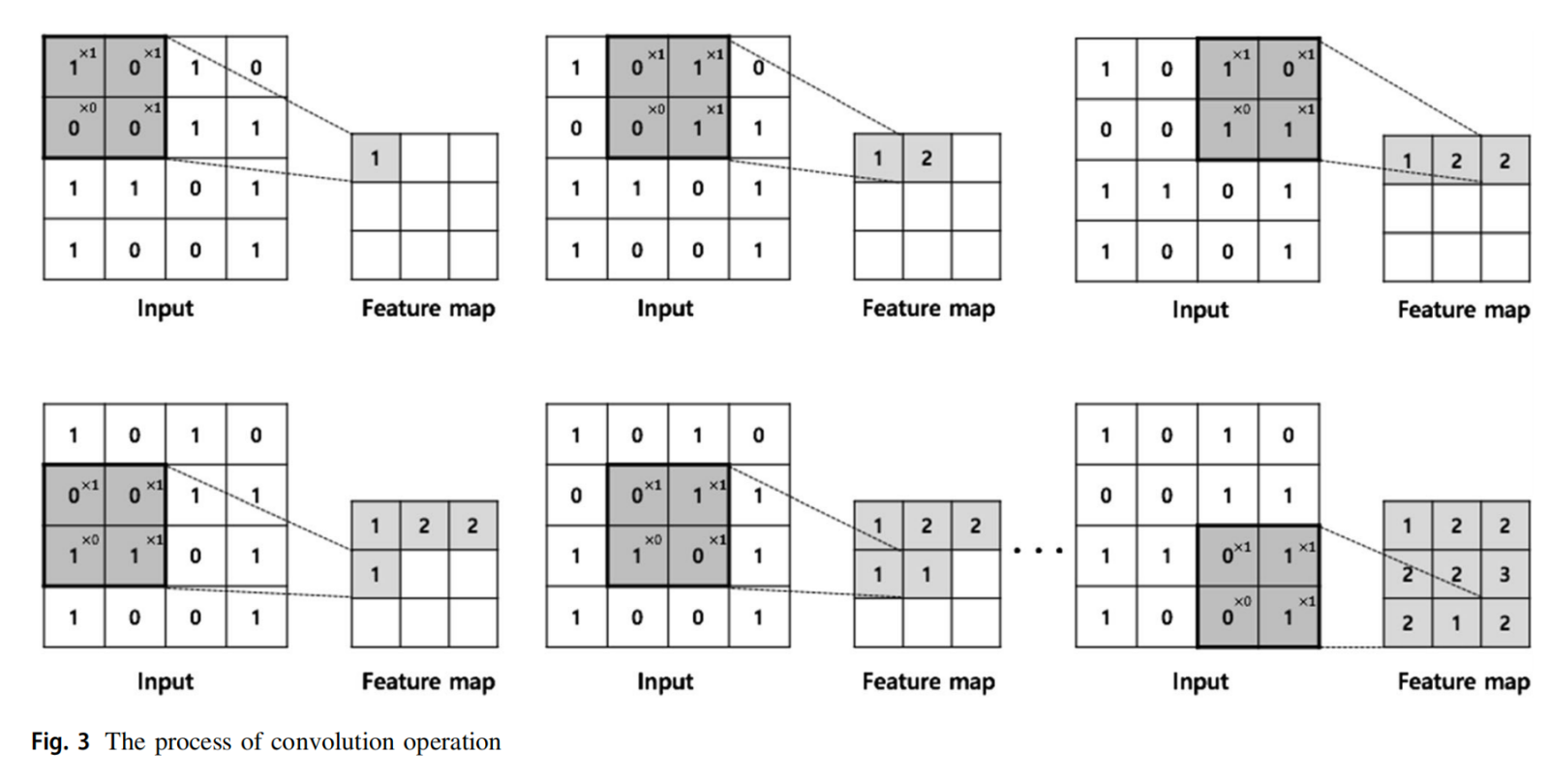
\*原始数据：【金山文档】 KS11-3 <https://kdocs.cn/l/cvg6Pca7llAP>

## **模型设计**

*The basic architecture of CNN*



*The process of convolution operation*



常规的CNN模型，往往由卷积层+池化层+全连接层组成。

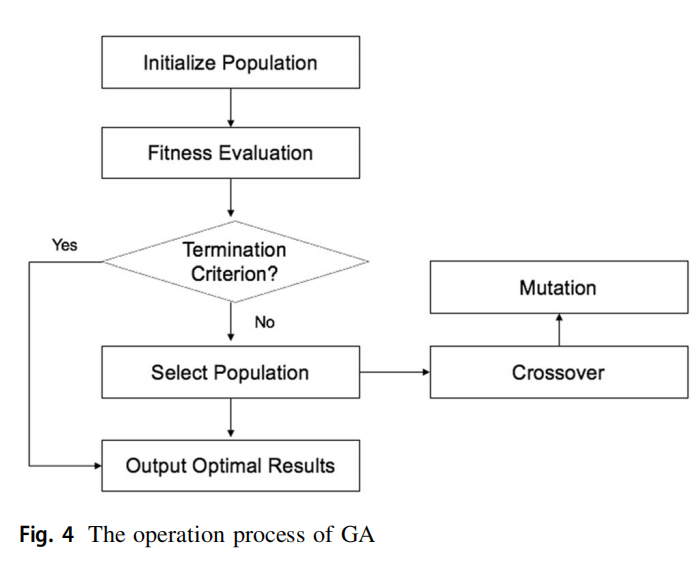
在本文中，作者使用了2个卷积层+1个池化层+2个全连接层，完成了整个模型的设计。

##### 卷积层设计

针对第二节提到的7个因子，在CNN卷积层分设7个channel。与一般常规CNN模型不同的是，本文未通过人为指定的方式，而是选用GA模型进行通道参数的遴选与优化，得出最优的网络结构。

###### Part 1 Calculate kernels of each channel by GA

* **GA计算流程：**



Step 1：种群初始化（Initialize Population）

Step 2：定义适应度函数（Fitness Function）并执行适应度评估（Fitness Evaluation）；

- Number of kernels in each convolutional layer: 1–63

- Kernel size of individual channels in the first convolutional layer: 1–31

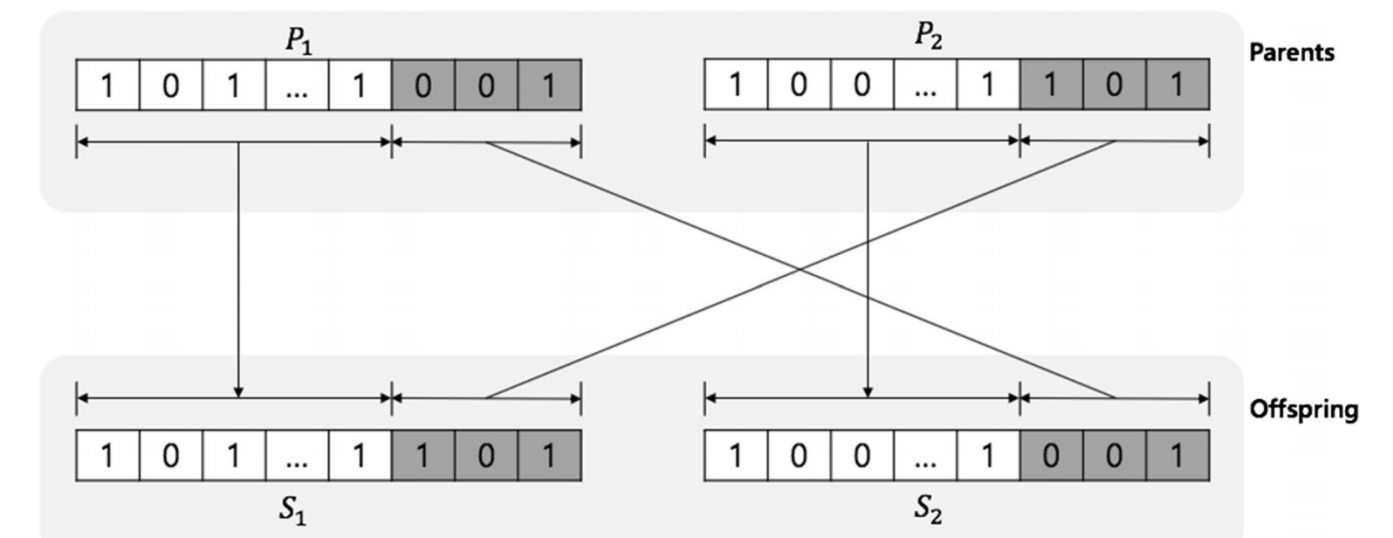
- Kernel size of individual channels in the second convolutional layer: 1–31

- Window size of the pooling layer: 1–31

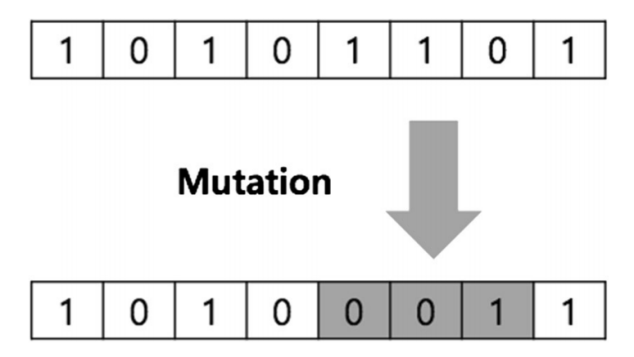
Step 3：Population是否满足目标条件函数（Fitness Function）之要求？

- 若已满足，则输出种群结果；

- 若未满足，则执行selection，从当前代中选择的双亲样本的部分染色体互换（Crossover），并创建一对新个体染色体并交换字符串中的指定部分；



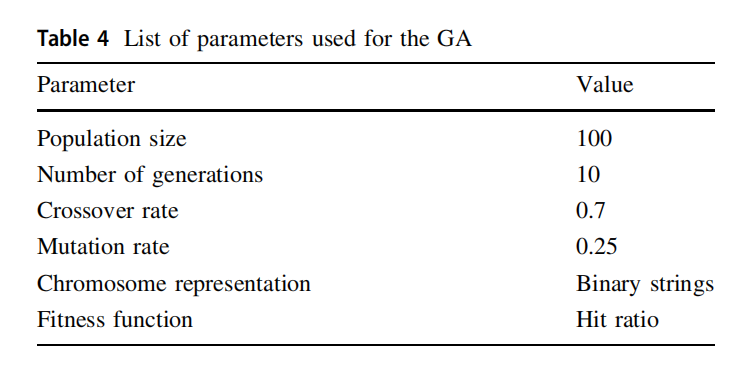
Step 4: 多次计算后，染色体产生突变；



Step 5: 再次校验是否满足Fitness Function之要求。

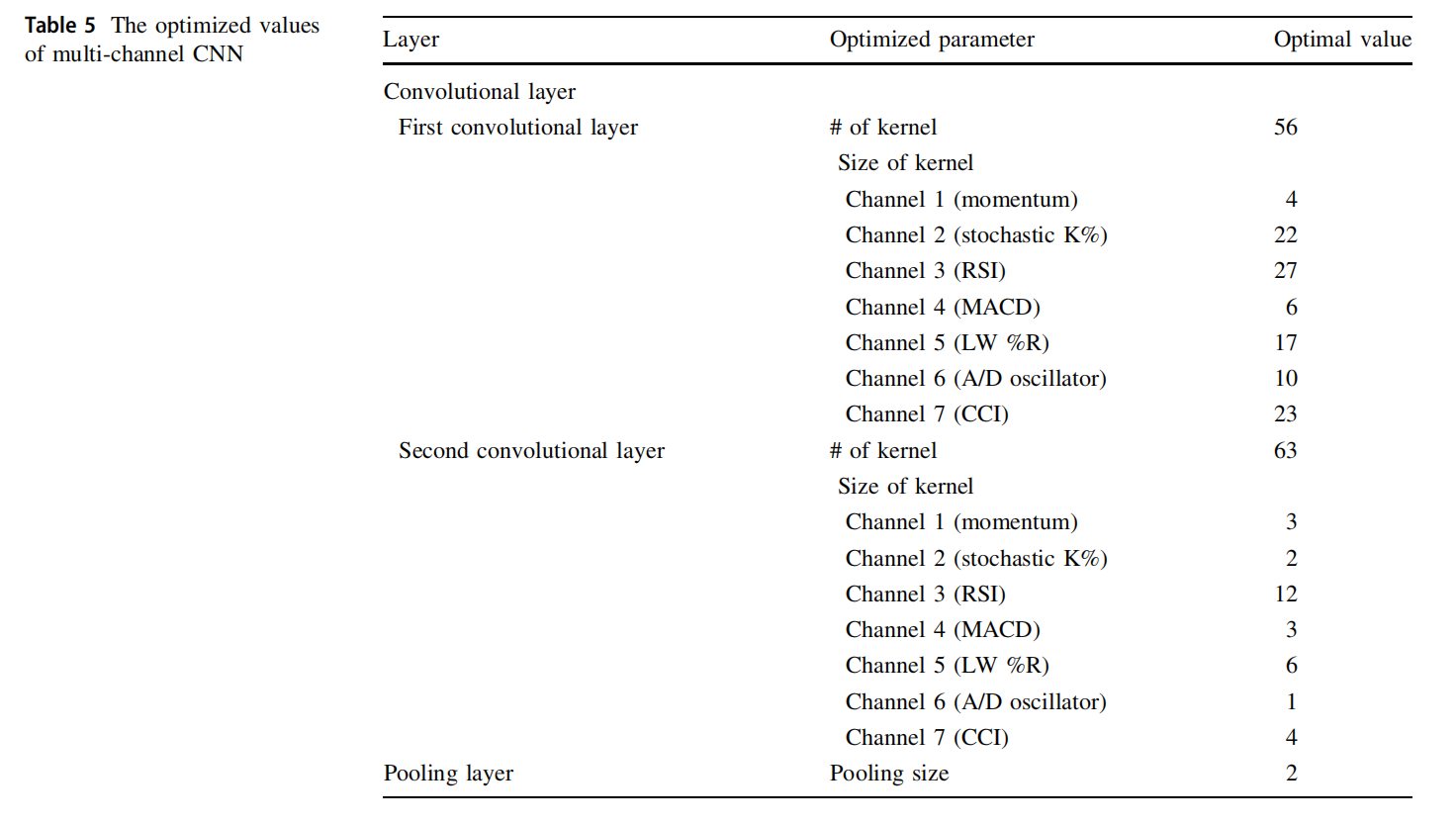
* **GA超参数：**

GA算法高度依赖超参数。参照以往研究，本次相关超参数取值如下：



* **运算结果：**

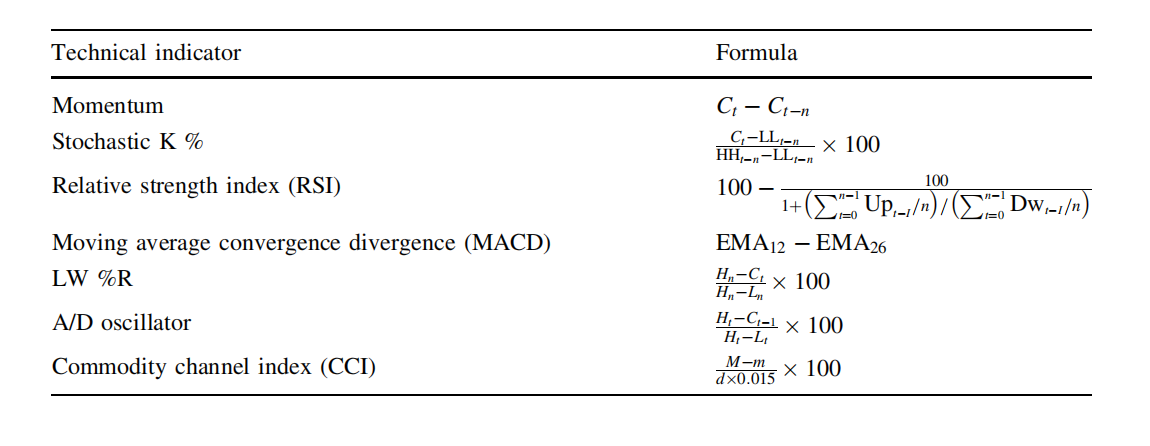
经过GA运算后，得出2个convolutional layer kernel个数如下：



###### Part 2 CNN输入变量准备

在本次研究中，共计人工指定并选用7个输入变量：

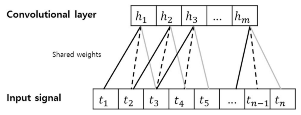
1. Momentum - 趋势指标
2. Stochastic K % - 超买超卖
3. RSI - 相对强弱指数
4. MACD - 异同移动平均线
5. LW %R - 资金流量指标
6. A/D oscillator - 震荡指标
7. CCD - 商品通道指数



###### Part 3 **CNN的数据输入：**

基于第一节所引用之dataset，拟采用80%作为训练集，20%作为验证集。

该模型的数据输入，采用从多元时间序列中分离出来的多个一维子序列。



##### **池化层设计**

经过GA运算后，期待得出两个pooling layer kernel个数如下：





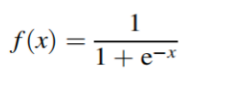
##### **全连通层设计**

本次研究中，我们构建了2个全连通层，第一个全连通层的数量为200个节点，第二个全连通层的数量为40个节点，均为人工指定。

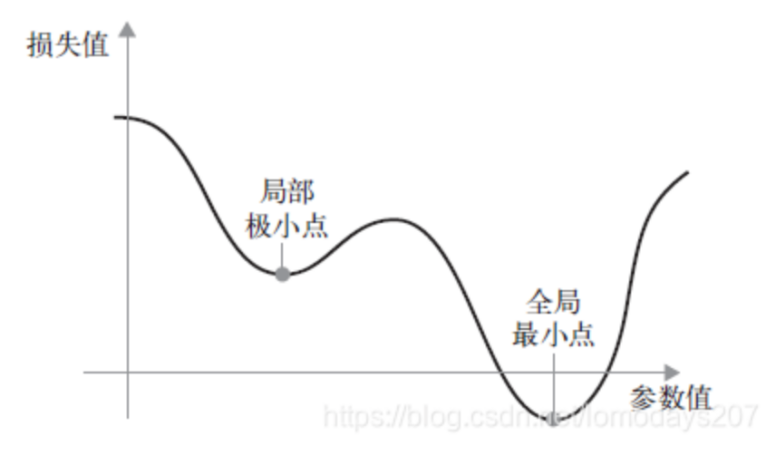
在第一层全连通层中，我们利用一维核来提取局部时间信息，并使用（ReLU）作为激活函数。带有输入x的ReLU函数可以表示如下：



在第二层全连通层中，为了进行二值分类，输出层采用Sigmiod Function作为激活函数：



此外，为了针对梯度下降时可能产生的局部最优问题，本文使用了Adam optimizer优化器。

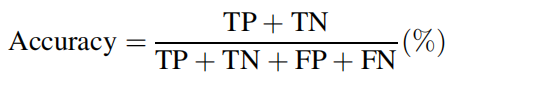


##### **数据输出**

该模型所期望的数据输出，为第二天股票指数的移动方向和输出值。

## **训练效果监测指标**

该模型的训练效果，将以Accuracy为观测指标，与ANN、CNN进行比较。Accuracy计算公式如下：



## **代码实现**

代码结构如下：

# Step 0: get data

# Step 1: Prepare the feats

# momentum

# stochastic K

# RSI

# macd

# LW %R

# A/D Oscillator

# CCI

# label

# Step 2: Normalize the Data

# Step 3: Split the Data

# Step 4: Make Trainable Dataset

# Step 5: Define the TimeSeriesCNN

# Input Layer

# Max Pooling Layer

# Fully Connected Layers

# Apply Convolutional Layers with ReLU Activation

# Apply Max Pooling Layer

# Flatten the output for the Fully Connected Layers

# Apply Fully Connected Layers with ReLU Activation

# Step 6: Training and Evaluation

# Instantiate the TimeSeriesCNN model

# Define loss function and optimizer

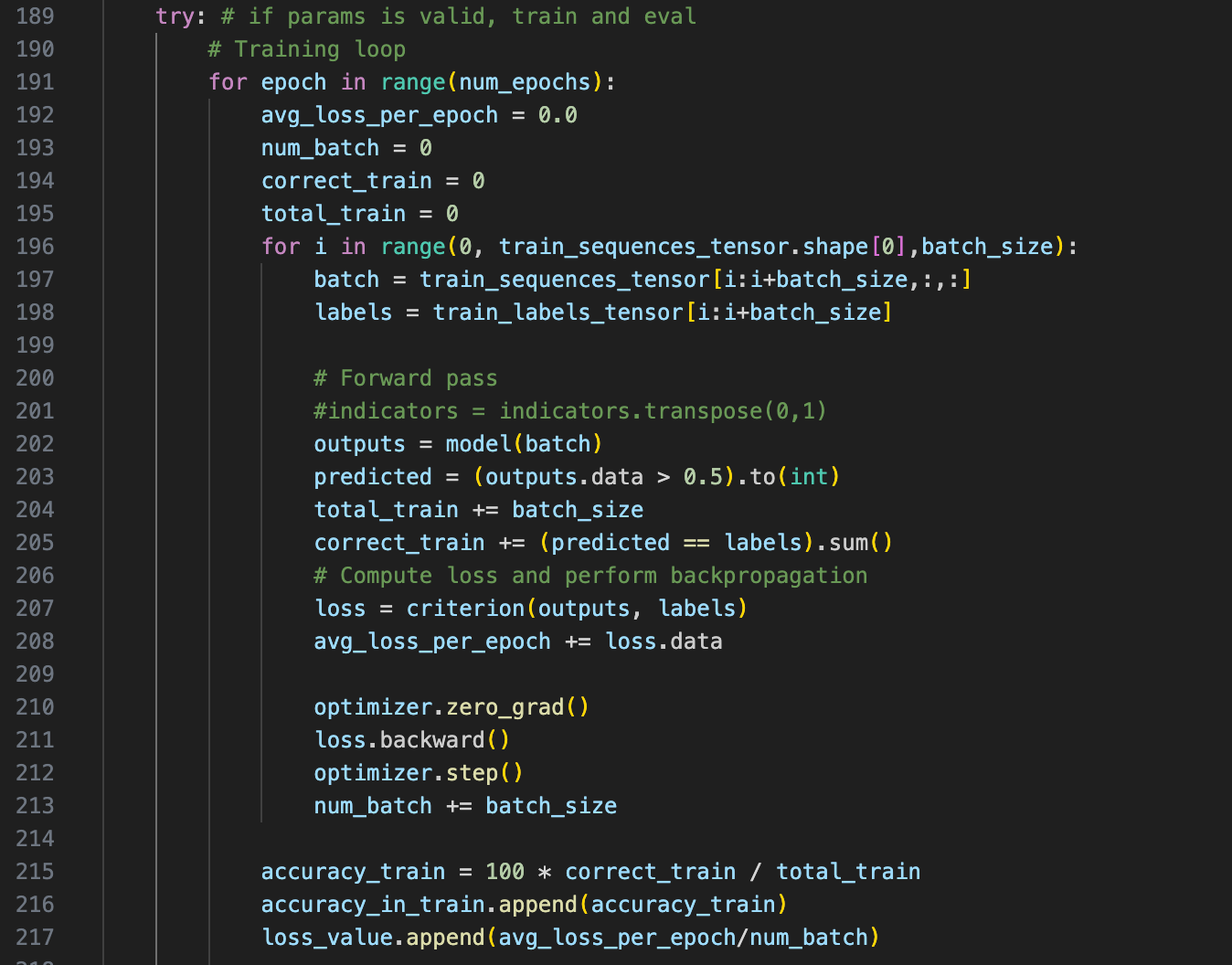
# step 7 define fitness func

# step 8 find best pop with GAs

# step 9 train and output best pop

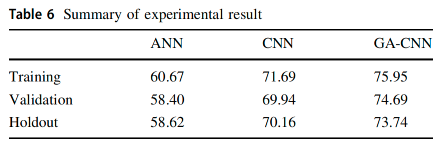
代码文档：

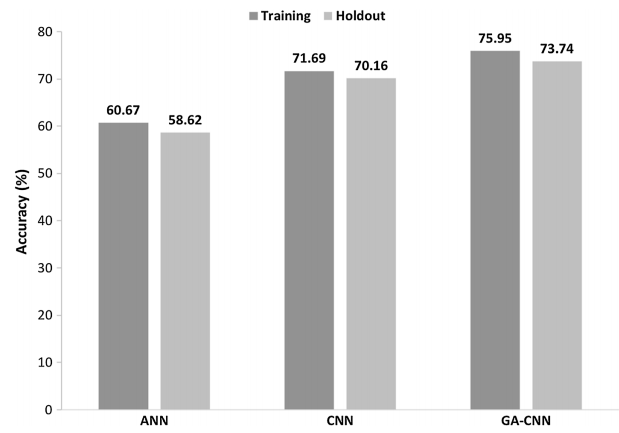




## **模型效果**

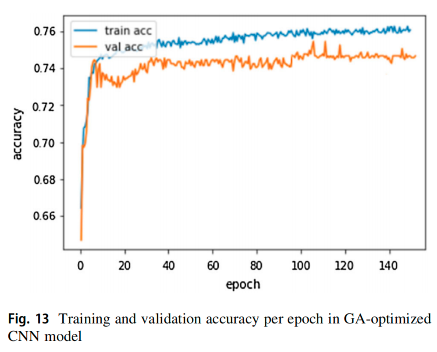
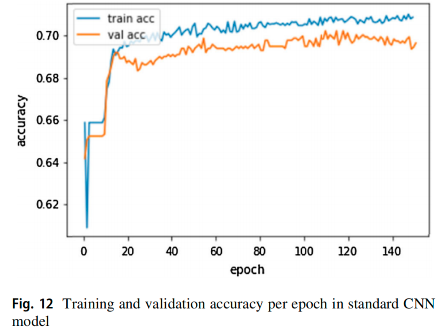
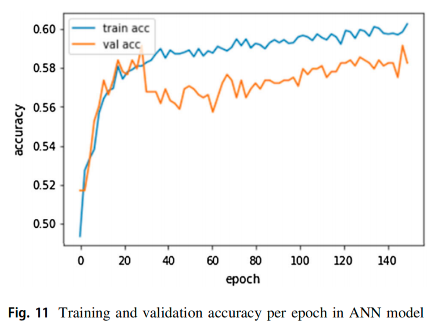
以Accuracy指标作为评价标准，对比ANN、CNN、GA-CNN三种模型的计算结果如下：





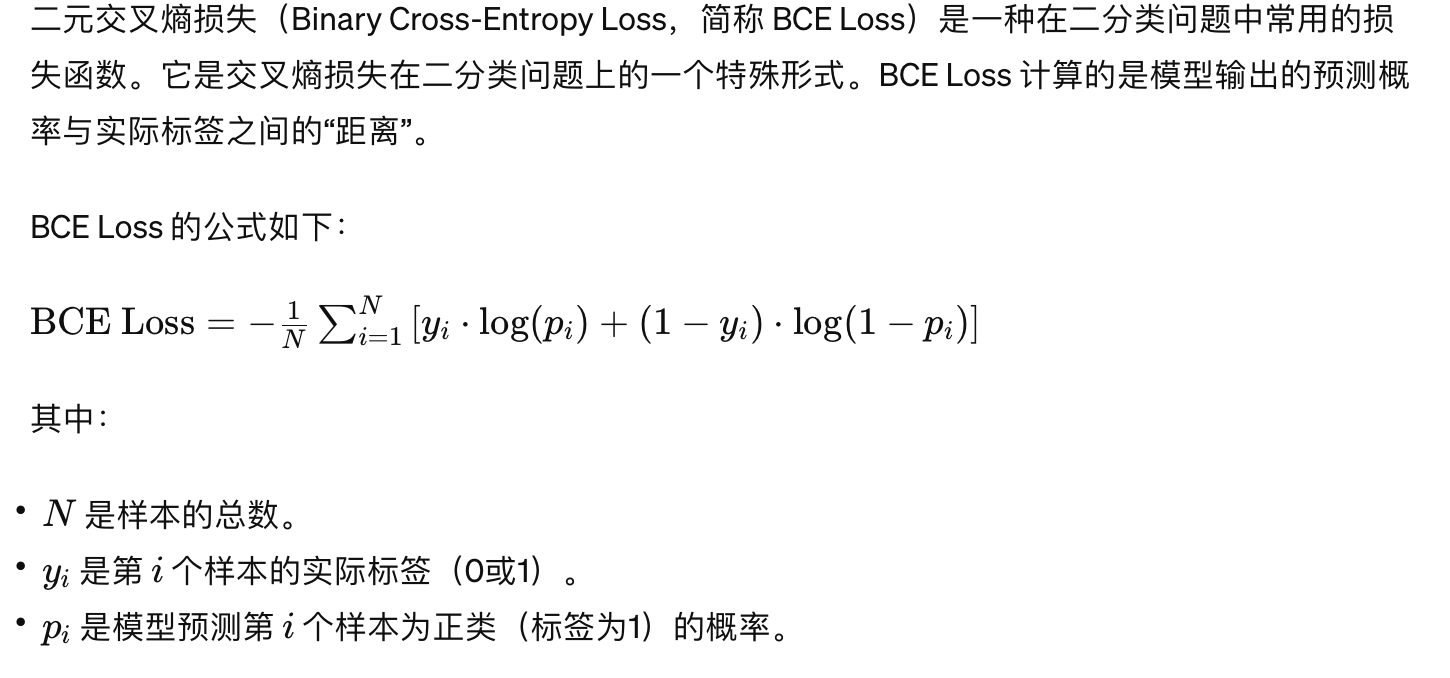
由如下3张图可看出，当epoch次数处于[40,150]区间时，accuracy值趋于稳定；

其中以GA-CNN模型的accuracy值最高，训练集稳定在0.76左右，验证集稳定在0.74左右。



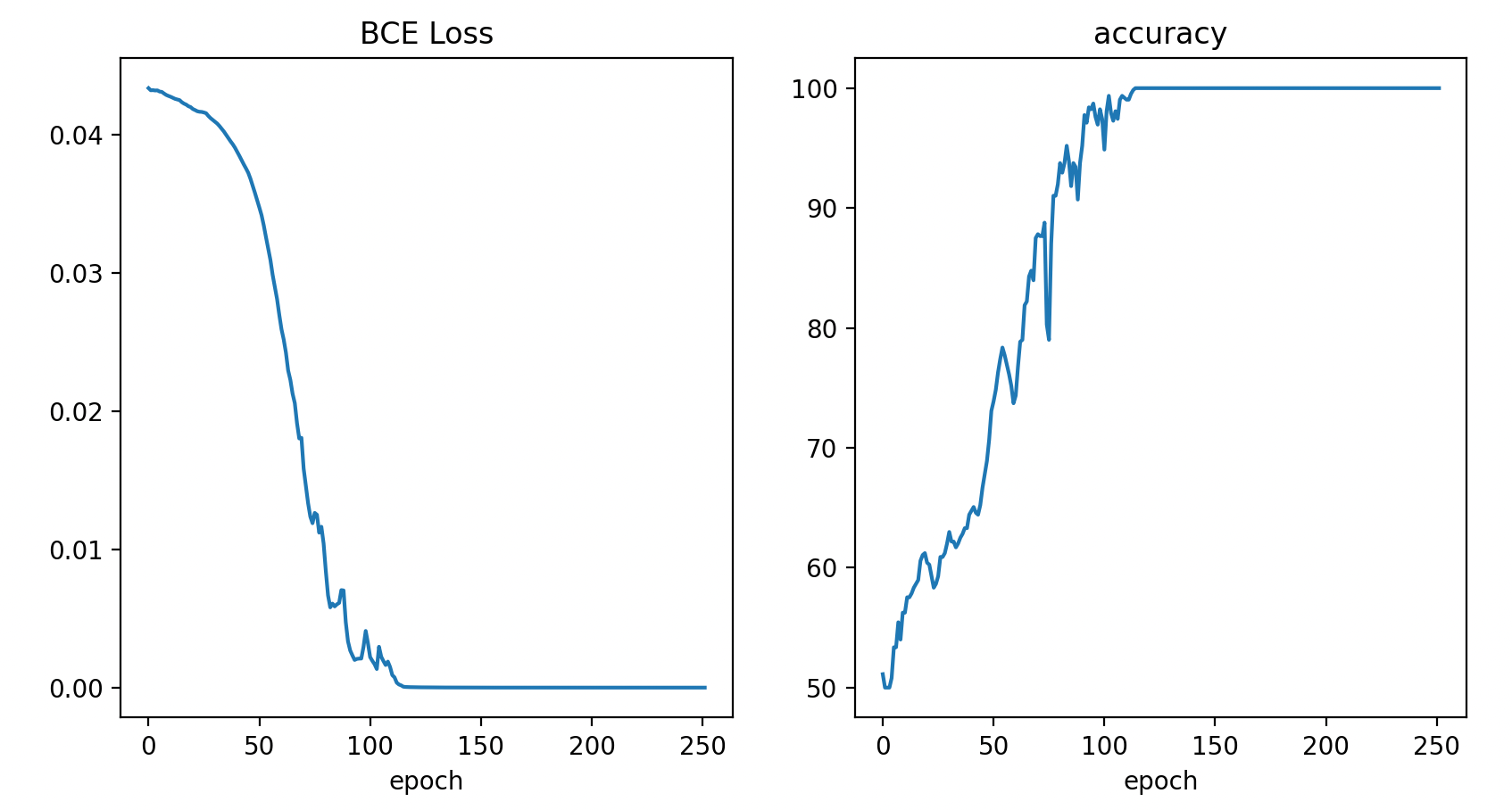
* **增补代码实现细节，共计4处：**

1. window size=64
2. num\_epochs = 1500
3. batch\_size = 8
4. MSE loss → BCE loss

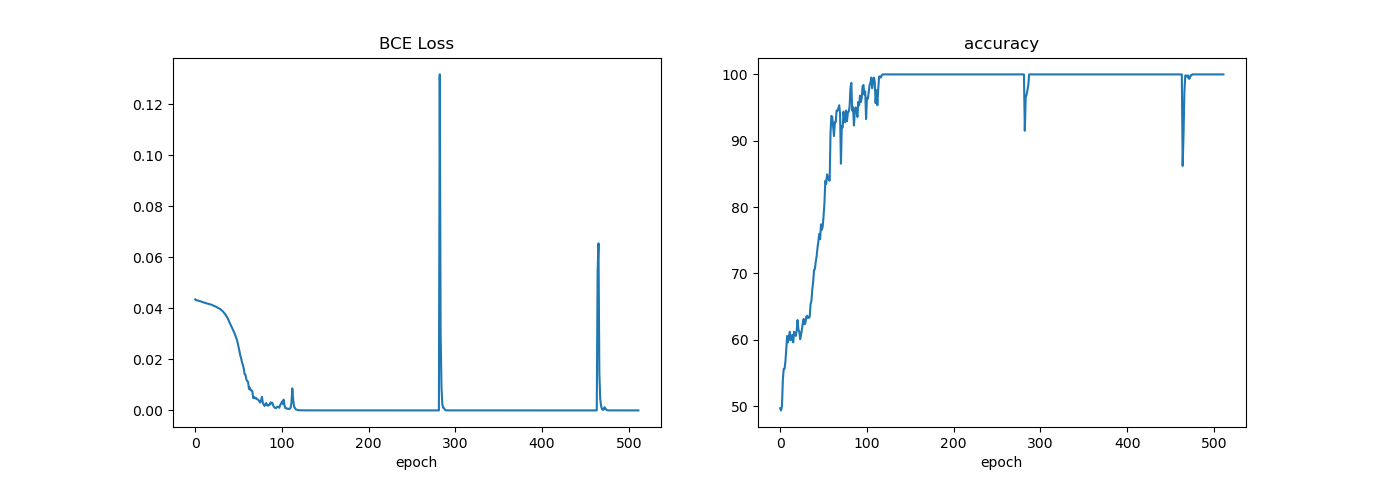


训练结果（11.13）：

When epoch = 252



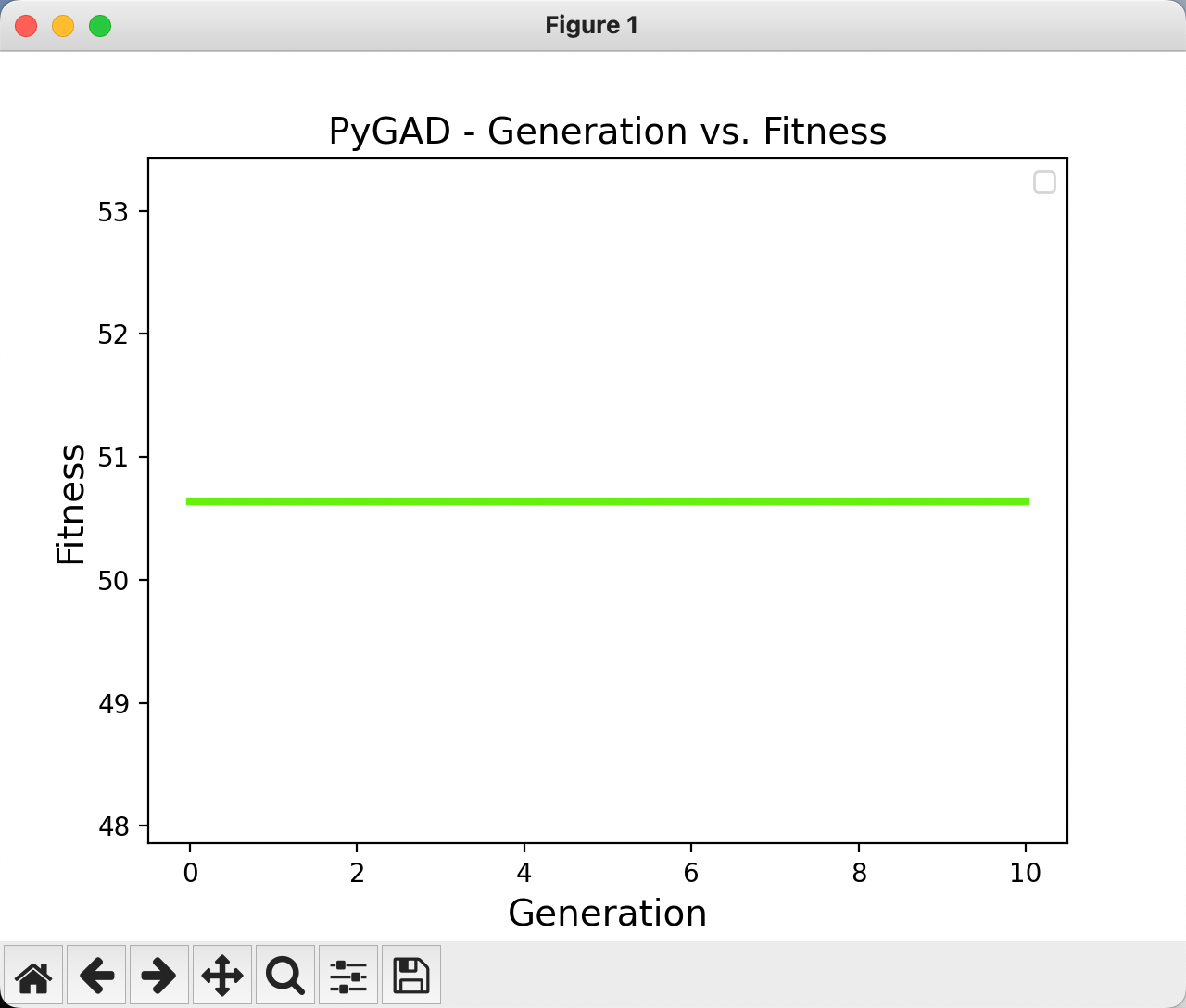
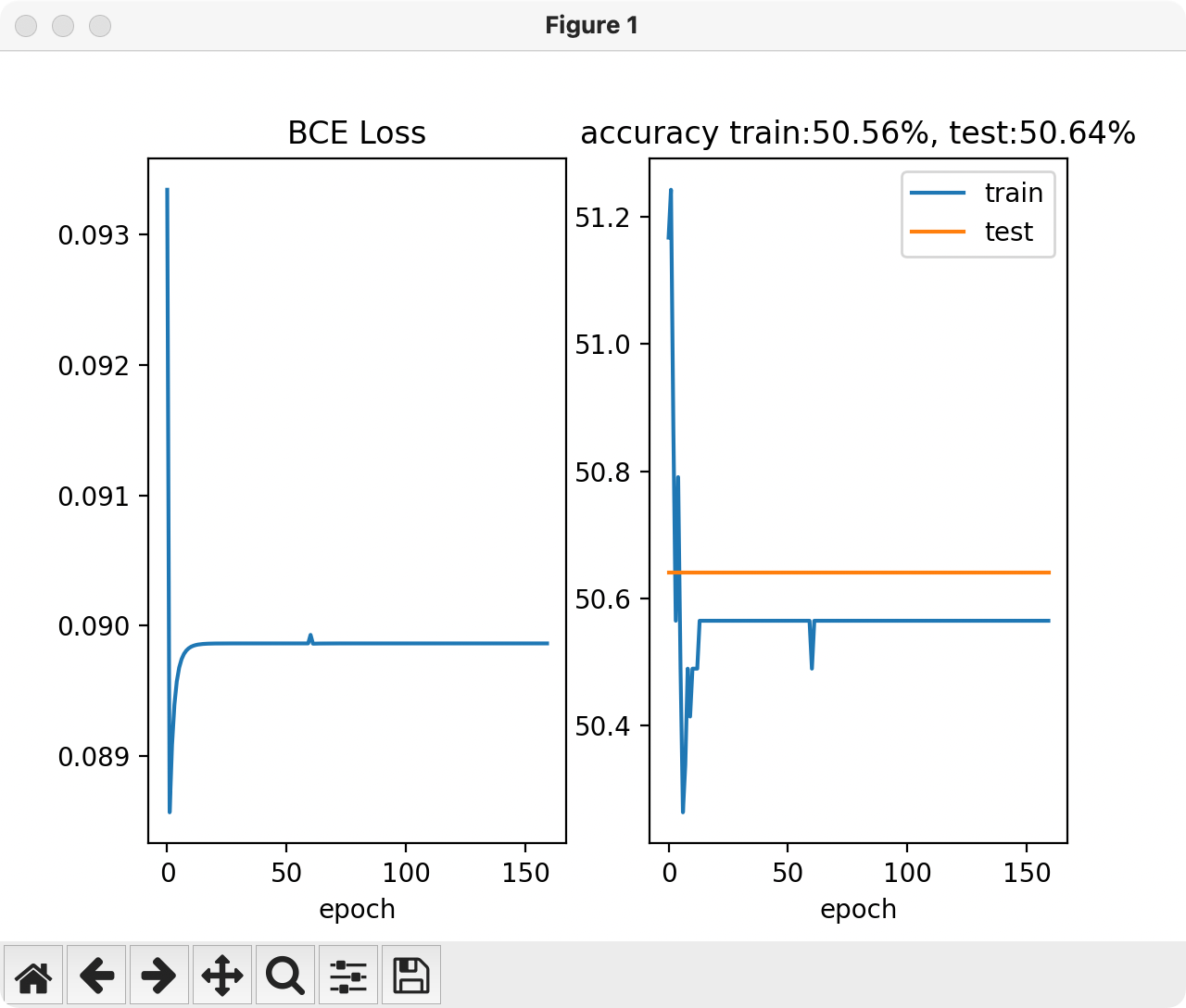
When epoch = 512



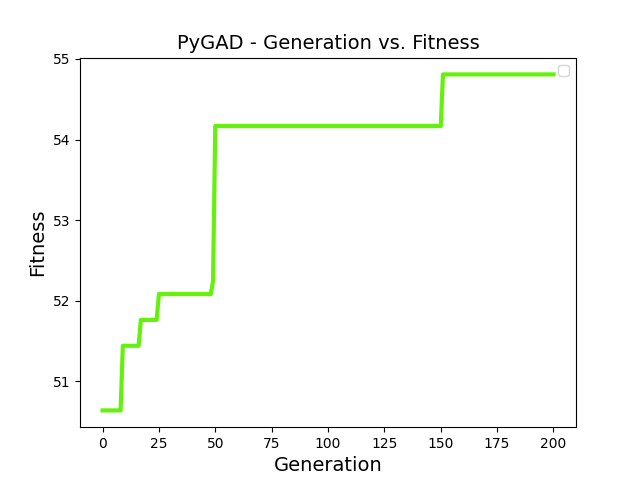
训练结果（12.1）：

1. 完成了GA部分的代码
2. window size=64
3. num\_generation=10/200/1500
4. num\_epochs = 160
5. batch\_size = 8

**When num\_generation=10,**

**When num\_generation=200,**

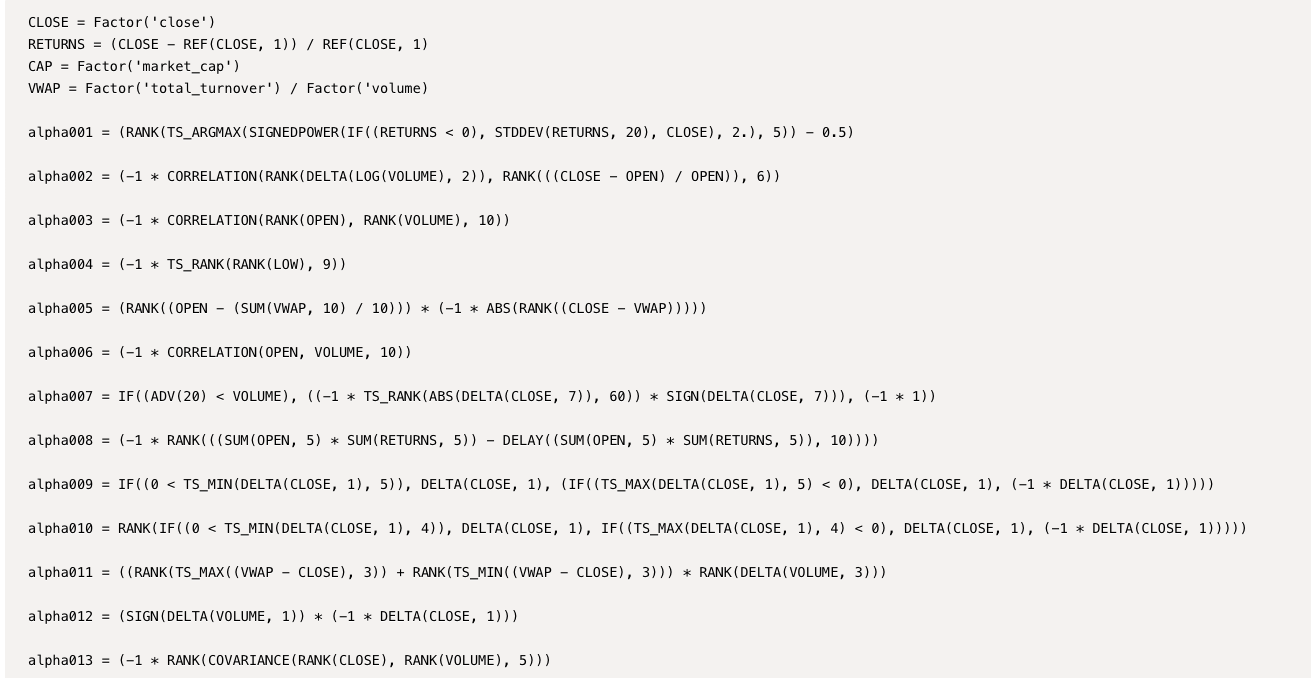


## **总结及建议**

几个拟进一步探讨&挖掘的方向：

1. 使用SSE index截至2023年数据，作进一步验证
2. 选择其他IC值更好的因子

<https://www.ricequant.com/doc/rqdata/python/factors-dictionary.html#alpha101-因子>



1. 除了指数涨跌趋势预测之外，该模型是否还可应用于单只股票的股价预测。