# 回归与分类报告

目录

[回归与分类报告 1](#_Toc519795543)

[动物电击回归 2](#_Toc519795544)

[采用决策树树回归 2](#_Toc519795545)

[采用 SVR回归 3](#_Toc519795546)

[采用LinearRegression 4](#_Toc519795547)

[采用随机森林回归RandomForestRegessor 5](#_Toc519795548)

[采用K邻近算法回归 5](#_Toc519795549)

[Indirect间接接触数据回归 6](#_Toc519795550)

[采用SVR回归 6](#_Toc519795551)

[采用决策树回归 7](#_Toc519795552)

[采用LinearRegressor 8](#_Toc519795553)

[采用GBRT梯度提升回归树 9](#_Toc519795554)

[采用KNN 10](#_Toc519795555)

[采用Adaboost回归 10](#_Toc519795556)

[Plants触电数据回归 11](#_Toc519795557)

[采用KNN算法回归 11](#_Toc519795558)

[采用SVR,rbf核函数 12](#_Toc519795559)

[采用随机森林 13](#_Toc519795560)

[MPLR神经网络回归 14](#_Toc519795561)

[keras自建神经网络 14](#_Toc519795562)

[改进思路 15](#_Toc519795563)

[结果 15](#_Toc519795564)

[三类数据分类问题 15](#_Toc519795565)

[基本做法 15](#_Toc519795566)

[改进方向 16](#_Toc519795567)

## 动物电击回归

代码详见：“animal.ipynb”

输入：5:钳形表100A

输出：4:钳形表100A电流

采用决策树树回归：

# DTR,数据不归一,前80

len\_X\_train = 109477

len\_X\_test = 36493

loss = 15930.519687184895

time cost = 0.1633707308440984 s

# DTR,数据归一,前80

len\_X\_train = 109477

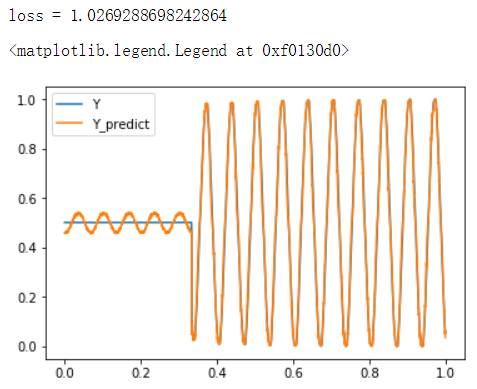
len\_X\_test = 36493

loss = 0.8123704877021091

time cost = 0.1633845337108475 s

--- 时间消耗上数据归一与不归一差不多，测试损失所有方法中最小

选取82号csv数据，作图



采用 SVR回归

# Rbf核函数，数据不归一，使用前40组(兔)数据

len\_X\_train = 54847

len\_X\_test = 18283

loss = 12473.184889567743

time cost = 351.9216612988851 s

# Rbf，数据归一，使用前40组数据

len\_X\_train = 54847

len\_X\_test = 18283

loss = 9.380495895967233

time cost = 0.21283560431783144 s

# Rbf，数据归一，使用前80组数据

len\_X\_train = 109477

len\_X\_test = 36493

loss = 7.484600671327897

time cost = 0.035597947265955554 s

# Linear，数据归一，前40组数据

len\_X\_train = 54847

len\_X\_test = 18283

loss = 7.542033740264803

time cost = 0.13149778800209333 s

# Linear，数据归一，前80组数据

len\_X\_train = 109477

len\_X\_test = 36493

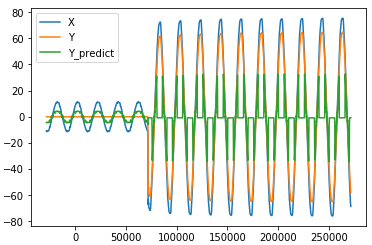
loss = 4.057021180083159

time cost = 0.03340753848874556 s

--- 数据归一后，学线性归一，习速度明显增快。

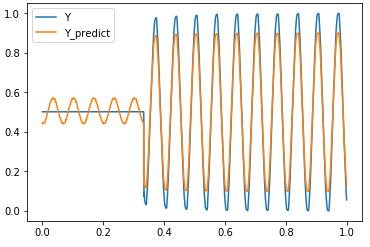
对于第一个训练器，数据不归一，只使用前40组数据训练，当用改训练器预测前40组数据时是可以得到较好结果，但预测后面羊的数据时，会发生很大的损失，因为后面数据自变量的最大值(80)比前40组的自变量最大值(20左右)大得多，当用此训练器去预测时，会发生截断

未归一化，rbf，前40组数据，82cvs测试：



--- 这里采用数据归一后，第二个训练器svr2相比于第一个（差别是是否归一化），因为采用线性归一化，所有自变量，因变量返回为[0,1];上面的问题将不会发生。并且训练速度极大地加快。并且数据归一化后，由于数据的重复性，相似性，采用前40组数据，与80组数据，其效果差别并不显著，当然后者的平均损失更小一些。

归一化，rbf，前40组数据，82cvs测试：



--- 这里用支持向量回归的方法，核函数取rbf高斯函数与线性函数时，可以看到，这里线性与rbf损失差不多。82号csv测试作图，与上面相同不再贴出。

采用LinearRegression

数据归一化，前80组数据

len\_X\_train = 109477

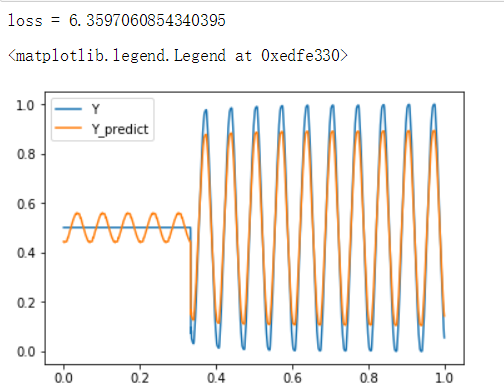
len\_X\_test = 36493

loss = 3.963673619098741

time cost = 0.005989382410234612 s

---这里，线性回归的方法优于采用SVR的回归方法，比决策树回归要差，训练速度很快。

选取82号csv数据为，作图



采用随机森林回归RandomForestRegessor

数据归一化，前80组数据

len\_X\_train = 109477

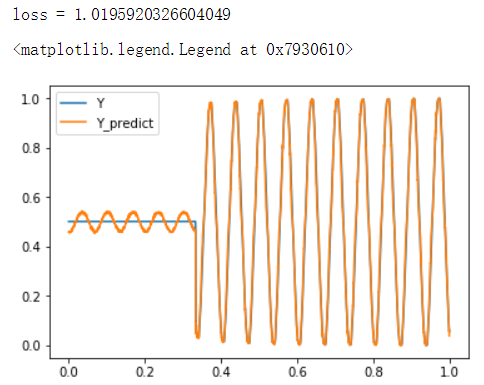
len\_X\_test = 36493

loss = 0.8020046417769066

time cost = 1.2872542912758804 s

---效果与决策树相似，非常好，速度快

选取82号csv数据，作图



采用K邻近算法回归

数据归一化，前80组数据

len\_X\_train = 109477

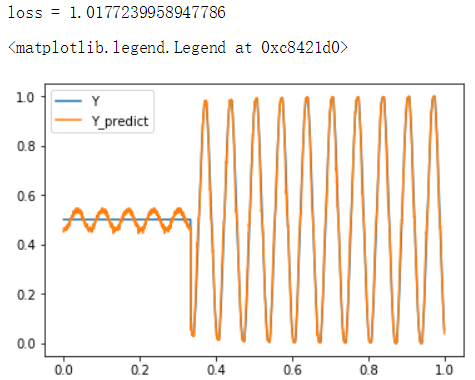
len\_X\_test = 36493

loss = 0.8579979600521066

time cost = 0.21835108830293848 s

---效果与决策树，随机森林，差不多，拟合效果好损失小，时间快

选取82号csv数据，作图测试



## Indirect间接接触数据回归

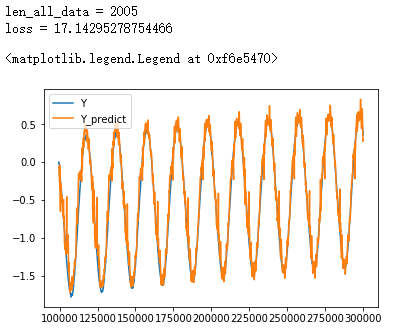
代码详见：“indirect.ipynb”

输入：1:交流电压180V 与 5:钳形表100A

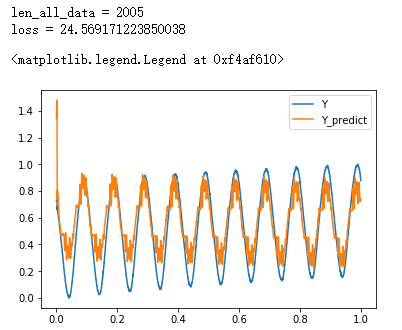
输出：4:钳形表100A电流

采用SVR回归

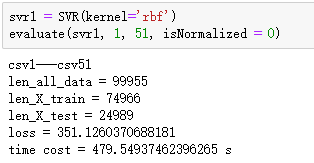
15-29组csv训练，以30号cvs测试，验证

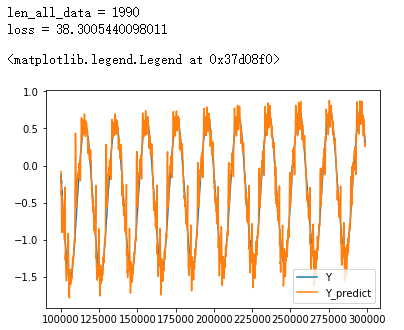


数据归一化后，损失大，拟合效果不佳，此后将不使用归一化数据：



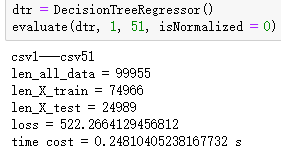
1-50训练， 59号csv数据测试

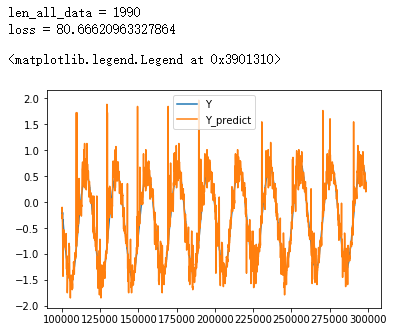




采用决策树回归

1-50组数据训练，59号csv数据作图测试





1-110组数据训练，112号数据作图测试

csv1---csv110

len\_all\_data = 217715

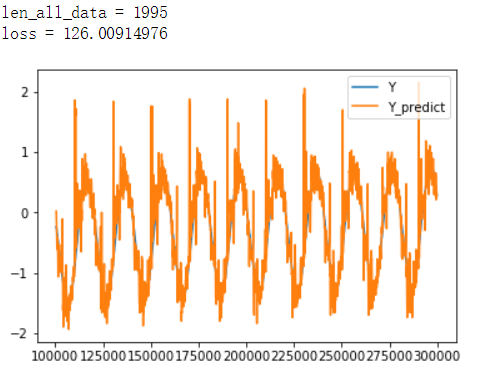
len\_X\_train = 163286

len\_X\_test = 54429

loss = 586.368274176

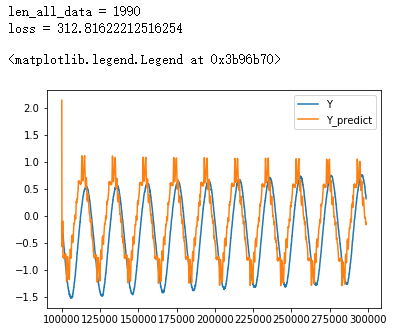
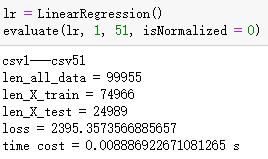
score = 0.860662170489

time cost = 0.4777800625233808 s

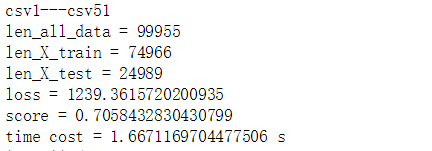


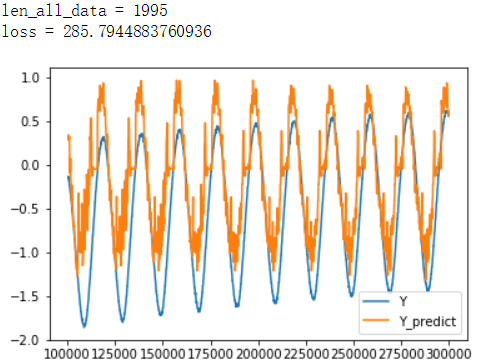
---此时决策树回归的效果不如SVR，并且，选取前50组数据训练，与选取前110组数据训练，效果差不多，表明，数据中有很多重复数据，可以采用特别的方法选取数据，如取奇数号的csv为训练数据，使其更具一般性。

采用LinearRegressor

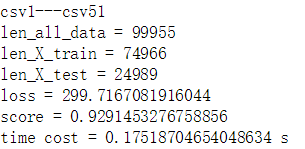


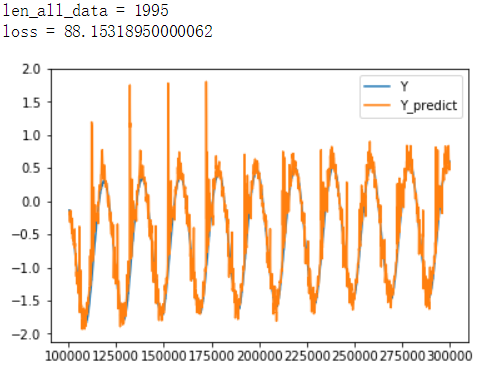
采用GBRT梯度提升回归树



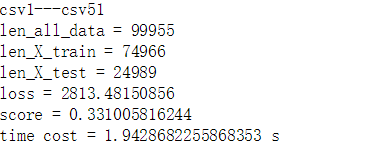


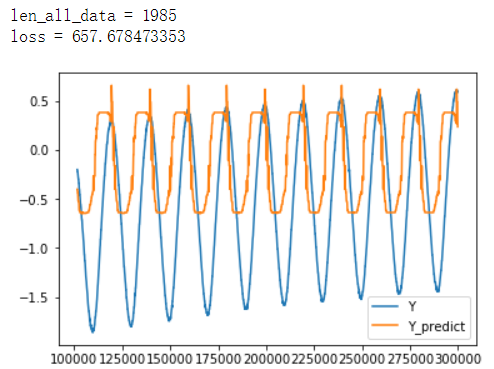
采用KNN





采用Adaboost回归





效果比较：SVR效果最好，时间最长>随机森林>KNN（拟合最好，但预测欠佳）>决策树回归>梯度提升GBR>线性回归>adaboost回归。（后三者效果很差）

## Plants触电数据回归

代码详见：“plants.ipynb”

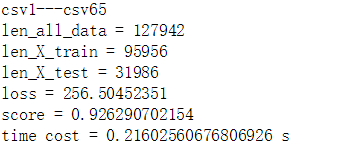
输入：1:交流电压180V 与 5:钳形表100A

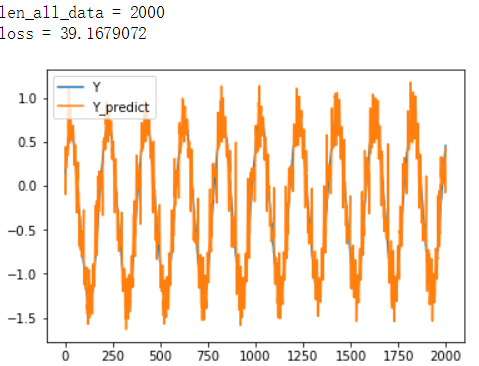
输出：4:钳形表100A电流

是否数据归一化：否

采用KNN算法回归

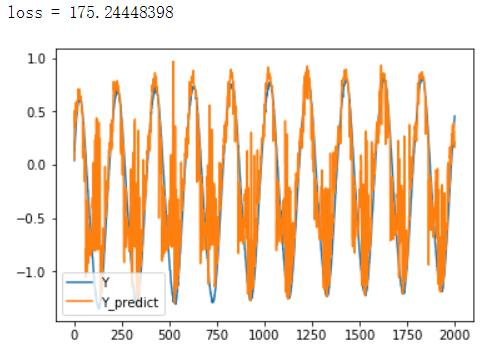
1-65号csv为训练数据，73号csv（随机找的）用来测试作图，





再加入相对时间序列到输入中，效果并未获得提升：

训练loss = 997;score=0.713



采用SVR,rbf核函数

csv1---csv65

len\_all\_data = 127942

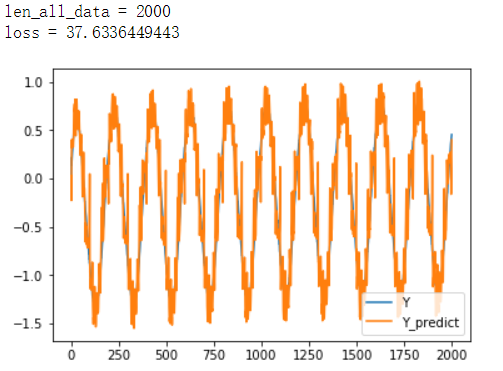
len\_X\_train = 95956

len\_X\_test = 31986

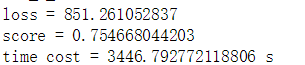
loss = 282.054477914

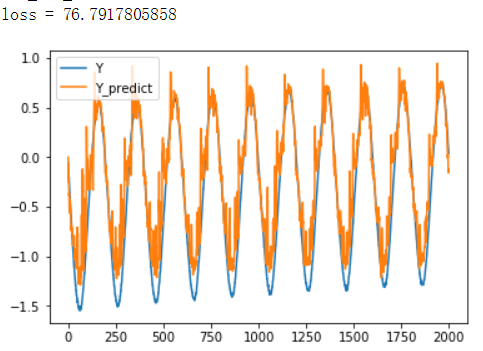
score = 0.918443571121

time cost = 545.4762642262281

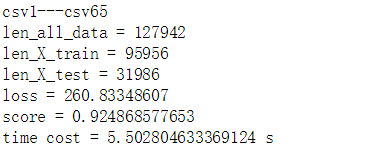


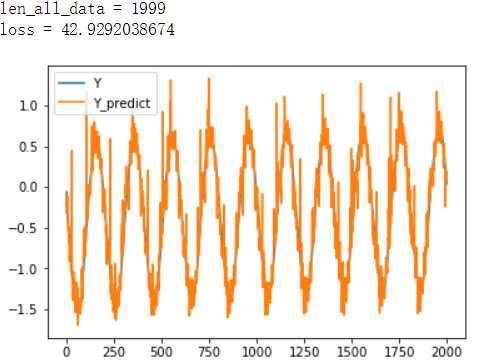
再加入相对时间到输入中，同上运行，效果更差了：



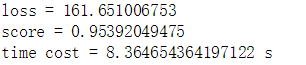


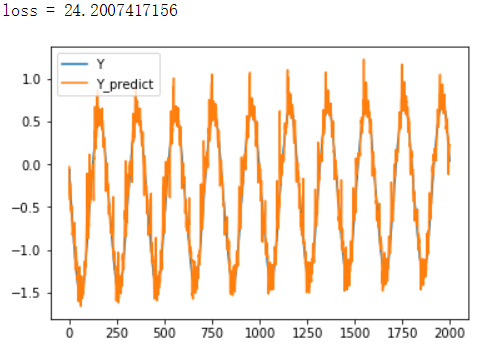
采用随机森林





再加入相对时间到输入中，同上运行：



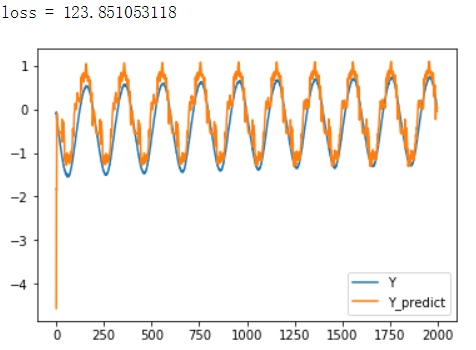
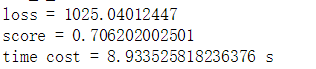


---此时，加入相对时间后，效果得到提升

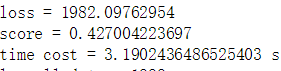
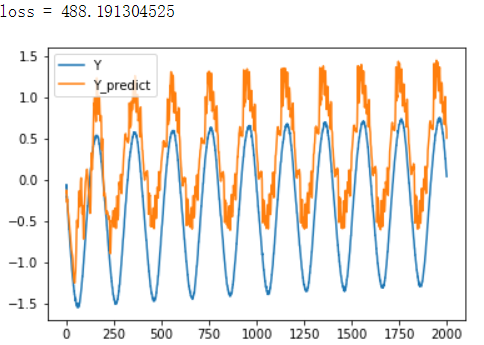
训练测试损失提升率，（260-161）/260=38.1%

MPLR神经网络回归

Csv1-csv65训练测试，csv73用来作图测试



再加入时间序列到输入中，效果更差了。

keras自建神经网络，效果很差

model = Sequential()

model.add(Dense(100, init='uniform', input\_dim=2))

model.add(Activation("tanh"))

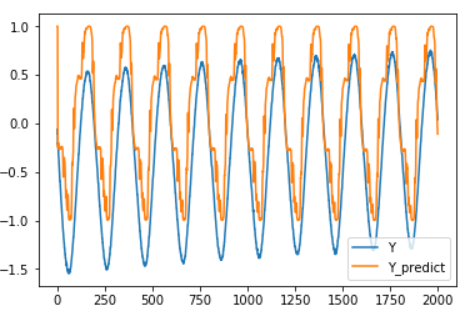
model.add(Dense(50))

model.add(Activation("tanh"))

model.add(Dense(1))

model.add(Activation("tanh"))

model.compile(loss='mean\_squared\_error', optimizer=SGD(lr=0.01, momentum=0.9, nesterov=True))



改进思路：需要自建简单的RNN网络，以解决时间序列，前后相关问题。当调用keras中SimpleRNN或LSTM时，会发生一下错误，还未解决。



考虑到时间相关的问题，也想尝试ARIMA（差分自回归模型）方法，可针对第二类indirect数据与第三类plants数据进行分析，得出三个参数（p, d, q）,

结果

第一未使用时间序列输入的随机森林回归，效果最佳。

随机森林(加入时间序列，loss1=161,loss2=24) > KNN( loss1=254, loss2=40)>SVR( 时间消耗太大，loss1=273 , loss2=30 ) > > 随机森林(未加入时间序列，loss1=282，loss2=37) >神经网络MPLR。

## 三类数据分类问题

代码详见：“classifier.ipynb”

基本做法：

对每个csv进行特征提取：两列数据5:钳形表100A为输入X 4:钳形表100A电流为输出Y。

对每个csv求出向量X的最大值，最小值，向量Y的最大值，最小值。

对每个csv，以X为输入，Y为输出，使用LinearRegressor做回归拟合，测试数据占0.3并求出其测试数据的损失值loss与得分score。

得到[X\_max,X\_min,Y\_max,Y\_min,cost,score]为一个csv数据的特征向量。

这样将得到85+120+75组特征数据，并分别给特征标注种类1,2,3。

最后以75+100+65组数据为训练集，其它数据为测试集。

用随机森林分类器分类（效果较好）标红为错误用例

训练得分score=1.0

测试结果：

[1 2 1 1 1 2 1 1 1 1]

[2 2 2 2 2 1 2 2 2 2 2 2 1 2 1 2 2 1 1 2]

[3 3 3 3 3 3 3 3 3 3]

用adaboost分类器分类（对第二种数据测试效果很差）

训练得分score=0.812

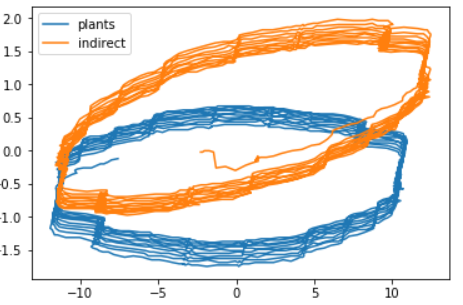
测试结果：

[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

[2, 2, 2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]

[3, 3, 3, 3, 3, 3, 3, 3, 3, 3]

对于indirect数据与plants数据，他们X与Y的分布图像大致如下



改进方向：主要是对对每个csv数据的特征提取方法的改进

1. 可以加入平均值与方差

2. 对于选择用来拟合数据的回归器，使用LinearRegressor，可以计算出其回归方乘的系数矩阵w与截距b，这两个也可以用来做为每个csv数据的特征。