问题一：

①第一问温度图

[mydata] = textread('input.txt');  
data = mydata  
l = 1; r = 2; cnt = 1;  
while l <= 114  
 while r < 114 && data(r, 1) < data(r + 1, 1)  
 r = r + 1;  
 end  
 temp = data(l:r, 1);  
 y1 = data(l:r, 2);  
 y2 = data(l:r, 3);  
 % figure(floor((cnt - 1) / 3) + 1)  
 if cnt > 1  
 figure(2)  
 subplot(4, 5, cnt - 1)  
 else  
 figure(1)  
 end  
 % myfigure = figure(cnt)  
 plot(temp, y1, 'r');  
 hold on  
 plot(temp, y2, 'b');  
 hold on  
 if cnt <= 14  
 str = sprintf("A" + num2str(cnt));  
 title(str)  
 else  
 str = sprintf("B" + num2str(cnt - 14));  
 title(str)  
 end  
 str\_t = sprintf('温度\n');  
 xlabel(str\_t)  
 % axis([250, 400, 0, 100])  
 if cnt == 1  
 legend('乙醇转化率', 'C4烯烃选择性')  
 end  
 % saveas(myfigure, num2str(cnt), 'png');  
 l = r + 1;  
 r = l + 1;  
 cnt = cnt + 1;  
end

②相似度计算

[input\_text] = textread('input.txt');  
input = input\_text;  
an = zeros(22, 1);  
for i = 1:22  
 an(i, 1) = sum(input(i, :).\*input(22, :));  
 an(i, 1) = an(i, 1) ./ sqrt(sum(input(i, :).^2));  
 an(i, 1) = an(i, 1) ./ sqrt(sum(input(22, :).^2));  
end  
an

问题二：

绘制3d图

①负载量与C4烯烃选择性关系图

import xlrd  
import random  
import pylab as mpl  
import matplotlib.pyplot as plt  
import numpy as np  
from matplotlib.font\_manager import FontProperties # 字体管理器  
  
# 设置汉字格式  
font = FontProperties(fname=r"consnerdi.ttf", size=12)  
x1 = [250, 275, 300, 325, 350]  
workbook = xlrd.open\_workbook(r'负载量&选择性.xlsx')  
sheet = workbook.sheet\_by\_index(0)  
x2 = [0.5, 1, 2, 5]  
fig = plt.figure()  
ax = fig.add\_subplot(projection = '3d')  
ys = np.array([0.5, 1, 2, 5])  
for i in range(0,5):  
 z = sheet.row\_values(i)  
 zs = np.array(z)  
 color = plt.cm.Set2(random.choice(range(plt.cm.Set2.N)))  
 ax.plot(ys, zs, x1[i], zdir='y', color='b', marker='o', alpha=0.8, label = str(x1[i]) + '℃')  
  
x1 = np.array(x1)  
for i in range(0,4):  
 z = sheet.col\_values(i)  
 x = x2[i]  
 zs = np.array(z)  
 color = plt.cm.Set2(random.choice(range(plt.cm.Set2.N)))  
 ax.plot(x1, zs, x, zdir='x', color='b', marker='o', alpha=0.8, label = str(x2[i]) + 'wt%')  
  
plt.title('Co负载量与C4烯烃选择性关系图', fontproperties = font)  
ax.set\_xlabel('Co负载量', fontproperties = font)  
ax.set\_zlabel('C4烯烃选择性', fontproperties = font)  
ax.set\_ylabel('温度', fontproperties = font)  
#ax.legend()  
plt.savefig("负载量&选择性.png")  
plt.show()

负载量与乙醇转化率关系图

import xlrd  
import random  
import pylab as mpl  
import matplotlib.pyplot as plt  
import numpy as np  
from matplotlib.font\_manager import FontProperties # 字体管理器  
  
# 设置汉字格式  
font = FontProperties(fname=r"consnerdi.ttf", size=12)  
x1 = [250, 275, 300, 325, 350]  
workbook = xlrd.open\_workbook(r'wt.xlsx')  
sheet = workbook.sheet\_by\_index(0)  
x2 = [0.5, 1, 2, 5]  
fig = plt.figure()  
ax = fig.add\_subplot(projection = '3d')  
ys = np.array([0.5, 1, 2, 5])  
for i in range(0,5):  
 z = sheet.row\_values(i)  
 zs = np.array(z)  
 color = plt.cm.Set2(random.choice(range(plt.cm.Set2.N)))  
 ax.plot(ys, zs, x1[i], zdir='y', color='b', marker='o', alpha=0.8, label = str(x1[i]) + '℃')  
  
x1 = np.array(x1)  
for i in range(0,4):  
 z = sheet.col\_values(i)  
 x = x2[i]  
 zs = np.array(z)  
 color = plt.cm.Set2(random.choice(range(plt.cm.Set2.N)))  
 ax.plot(x1, zs, x, zdir='x', color='b', marker='o', alpha=0.8, label = str(x2[i]) + 'wt%')  
  
plt.title('Co负载量与乙醇转化率关系图', fontproperties = font)  
ax.set\_xlabel('Co负载量', fontproperties = font)  
ax.set\_zlabel('乙醇转化率', fontproperties = font)  
ax.set\_ylabel('温度', fontproperties = font)  
#ax.legend()  
plt.savefig("温度&wt.png")  
plt.show()

②装料比与乙醇转化率关系图

#A1 A2 A4 A6  
import xlrd  
import random  
import pylab as mpl  
import matplotlib.pyplot as plt  
import numpy as np  
from matplotlib.font\_manager import FontProperties # 字体管理器  
  
# 设置汉字格式  
font = FontProperties(fname=r"consnerdi.ttf", size=12)  
workbook = xlrd.open\_workbook(r'装料比&转化率.xlsx')  
sheet = workbook.sheet\_by\_index(0)  
x1 = [250, 275, 300, 350, 400]  
x2 = [0, 0.5556, 1, 2.0303]  
fig = plt.figure()  
ax = fig.add\_subplot(projection = '3d')  
  
for i in range(0, 5):  
 z = sheet.row\_values(i)  
 x = x1[i]  
 xs = np.array(x2)  
 zs = np.array(z)  
 ax.plot(xs, zs, x, zdir='y', color='b', marker='o', alpha=0.8)  
  
for i in range(0, 4):  
 z = sheet.col\_values(i)  
 x = x2[i]  
 xs = np.array(x1)  
 zs = np.array(z)  
 ax.plot(xs, zs, x, zdir='x', color='b', marker='o', alpha=0.8)  
  
plt.title('HAP和Co/SiO2装料比与乙醇转化率关系图', fontproperties = font)  
ax.set\_xlabel('HAP和Co/SiO2装料比', fontproperties = font)  
ax.set\_zlabel('乙醇转化率', fontproperties = font)  
ax.set\_ylabel('温度', fontproperties = font)  
plt.savefig("装料比1.png")  
plt.show()

装料比与C4烯烃选择性关系图

#A1 A2 A4 A6  
import xlrd  
import random  
import pylab as mpl  
import matplotlib.pyplot as plt  
import numpy as np  
from matplotlib.font\_manager import FontProperties # 字体管理器  
  
# 设置汉字格式  
font = FontProperties(fname=r"consnerdi.ttf", size=12)  
workbook = xlrd.open\_workbook(r'装料比&选择性.xlsx')  
sheet = workbook.sheet\_by\_index(0)  
x1 = [250, 275, 300, 350, 400]  
x2 = [0, 0.5556, 1, 2.0303]  
fig = plt.figure()  
ax = fig.add\_subplot(projection = '3d')  
  
for i in range(0, 5):  
 z = sheet.row\_values(i)  
 x = x1[i]  
 xs = np.array(x2)  
 zs = np.array(z)  
 ax.plot(xs, zs, x, zdir='y', color='b', marker='o', alpha=0.8)  
  
for i in range(0, 4):  
 z = sheet.col\_values(i)  
 x = x2[i]  
 xs = np.array(x1)  
 zs = np.array(z)  
 ax.plot(xs, zs, x, zdir='x', color='b', marker='o', alpha=0.8)  
  
plt.title('HAP和Co/SiO2装料比与C4烯烃选择性关系图', fontproperties = font)  
ax.set\_xlabel('HAP和Co/SiO2装料比', fontproperties = font)  
ax.set\_zlabel('C4烯烃选择性', fontproperties = font)  
ax.set\_ylabel('温度', fontproperties = font)  
plt.savefig("装料比2.png")  
plt.show()

③乙醇浓度与乙醇转化率关系图

#A1 A2 A4 A6  
import xlrd  
import random  
import pylab as mpl  
import matplotlib.pyplot as plt  
import numpy as np  
from matplotlib.font\_manager import FontProperties # 字体管理器  
  
# 设置汉字格式  
font = FontProperties(fname=r"consnerdi.ttf", size=12)  
workbook = xlrd.open\_workbook(r'乙醇浓度&转化率.xlsx')  
sheet = workbook.sheet\_by\_index(0)  
x1 = [250, 275, 300, 350, 400]  
x2 = [0.3, 0.9, 1.68, 2.1]  
x2\_ = [1.68, 2.1]  
fig = plt.figure()  
ax = fig.add\_subplot(projection = '3d')  
  
for i in range(0, 5):  
 z = sheet.row\_values(i)  
 x = x1[i]  
 xs = np.array(x2)  
 zs = np.array(z)  
 ax.plot(xs, zs, x, zdir='y', color='b', marker='o', alpha=0.8)  
  
workbook\_ = xlrd.open\_workbook(r'乙醇B.xlsx')  
sheet\_ = workbook\_.sheet\_by\_index(0)  
for i in range(0, 5):  
 z = sheet\_.row\_values(i)  
 x = x1[i]  
 xs = np.array(x2\_)  
 zs = np.array(z)  
 ax.plot(xs, zs, x, zdir='y', color='r', marker='o', alpha=0.8)  
  
l1 = []  
l2 = []  
for i in range(0, 3):  
 z = sheet.col\_values(i)  
 x = x2[i]  
 xs = np.array(x1)  
 zs = np.array(z)  
 ax.plot(xs, zs, x, zdir='x', color='b', marker='o', alpha=0.8)  
  
z = sheet.col\_values(3)  
x = x2[3]  
xs = np.array(x1)  
zs = np.array(z)  
ax.plot(xs, zs, x, zdir='x', color='b', marker='o', alpha=0.8, label = 'A装填方式')  
  
  
for i in range(0, 1):  
 z = sheet\_.col\_values(i)  
 x = x2\_[i]  
 xs = np.array(x1)  
 zs = np.array(z)  
 ax.plot(xs, zs, x, zdir='x', color='r', marker='o', alpha=0.8)  
  
z = sheet\_.col\_values(1)  
x = x2\_[1]  
xs = np.array(x1)  
zs = np.array(z)  
ax.plot(xs, zs, x, zdir='x', color='r', marker='o', alpha=0.8, label = 'B装填方式')  
  
plt.title('乙醇浓度与乙醇转化率关系图', fontproperties = font)  
ax.set\_xlabel('乙醇浓度', fontproperties = font)  
ax.set\_zlabel('乙醇转化率', fontproperties = font)  
ax.set\_ylabel('温度', fontproperties = font)  
plt.legend(loc = 'upper left', prop = font)  
#ax.legend(loc = 'upper left')  
plt.savefig("浓度1.png")  
plt.show()

乙醇浓度与C4烯烃选择性关系图

#A1 A2 A4 A6  
import xlrd  
import random  
import pylab as mpl  
import matplotlib.pyplot as plt  
import numpy as np  
from matplotlib.font\_manager import FontProperties # 字体管理器  
  
# 设置汉字格式  
font = FontProperties(fname=r"consnerdi.ttf", size=12)  
workbook = xlrd.open\_workbook(r'乙醇浓度&选择性.xlsx')  
sheet = workbook.sheet\_by\_index(0)  
x1 = [250, 275, 300, 350, 400]  
x2 = [0.3, 0.9, 1.68, 2.1]  
x2\_ = [1.68, 2.1]  
fig = plt.figure()  
ax = fig.add\_subplot(projection = '3d')  
  
for i in range(0, 5):  
 z = sheet.row\_values(i)  
 x = x1[i]  
 xs = np.array(x2)  
 zs = np.array(z)  
 ax.plot(xs, zs, x, zdir='y', color='b', marker='o', alpha=0.8)  
  
workbook\_ = xlrd.open\_workbook(r'乙醇浓度&选择性 B.xlsx')  
sheet\_ = workbook\_.sheet\_by\_index(0)  
for i in range(0, 5):  
 z = sheet\_.row\_values(i)  
 x = x1[i]  
 xs = np.array(x2\_)  
 zs = np.array(z)  
 ax.plot(xs, zs, x, zdir='y', color='r', marker='o', alpha=0.8)  
  
l1 = []  
l2 = []  
for i in range(0, 3):  
 z = sheet.col\_values(i)  
 x = x2[i]  
 xs = np.array(x1)  
 zs = np.array(z)  
 ax.plot(xs, zs, x, zdir='x', color='b', marker='o', alpha=0.8)  
  
z = sheet.col\_values(3)  
x = x2[3]  
xs = np.array(x1)  
zs = np.array(z)  
ax.plot(xs, zs, x, zdir='x', color='b', marker='o', alpha=0.8, label = 'A装填方式')  
  
for i in range(0, 1):  
 z = sheet\_.col\_values(i)  
 x = x2\_[i]  
 xs = np.array(x1)  
 zs = np.array(z)  
 ax.plot(xs, zs, x, zdir='x', color='r', marker='o', alpha=0.8)  
  
z = sheet\_.col\_values(1)  
x = x2\_[1]  
xs = np.array(x1)  
zs = np.array(z)  
ax.plot(xs, zs, x, zdir='x', color='r', marker='o', alpha=0.8, label = 'B装填方式')  
  
plt.title('乙醇浓度与C4烯烃选择性关系图', fontproperties = font)  
ax.set\_xlabel('乙醇浓度', fontproperties = font)  
ax.set\_zlabel('C4烯烃选择性', fontproperties = font)  
ax.set\_ylabel('温度', fontproperties = font)  
plt.legend(loc = 'upper left', prop = font)  
#ax.legend(loc = 'upper left')  
plt.savefig("浓度2.png")  
plt.show()

乙醇浓度与C4烯烃收率关系图

#A1 A2 A4 A6  
import xlrd  
import random  
import pylab as mpl  
import matplotlib.pyplot as plt  
import numpy as np  
from matplotlib.font\_manager import FontProperties # 字体管理器  
  
# 设置汉字格式  
font = FontProperties(fname=r"consnerdi.ttf", size=12)  
workbook = xlrd.open\_workbook(r'乙醇收率A.xlsx')  
sheet = workbook.sheet\_by\_index(0)  
x1 = [250, 275, 300, 350, 400]  
x2 = [0.3, 0.9, 1.68, 2.1]  
x2\_ = [1.68, 2.1]  
fig = plt.figure()  
ax = fig.add\_subplot(projection = '3d')  
  
for i in range(0, 5):  
 z = sheet.row\_values(i)  
 x = x1[i]  
 xs = np.array(x2)  
 zs = np.array(z)  
 ax.plot(xs, zs, x, zdir='y', color='b', marker='o', alpha=0.8)  
  
workbook\_ = xlrd.open\_workbook(r'乙醇收率B.xlsx')  
sheet\_ = workbook\_.sheet\_by\_index(0)  
for i in range(0, 5):  
 z = sheet\_.row\_values(i)  
 x = x1[i]  
 xs = np.array(x2\_)  
 zs = np.array(z)  
 ax.plot(xs, zs, x, zdir='y', color='r', marker='o', alpha=0.8)  
  
l1 = []  
l2 = []  
for i in range(0, 3):  
 z = sheet.col\_values(i)  
 x = x2[i]  
 xs = np.array(x1)  
 zs = np.array(z)  
 ax.plot(xs, zs, x, zdir='x', color='b', marker='o', alpha=0.8)  
  
z = sheet.col\_values(3)  
x = x2[3]  
xs = np.array(x1)  
zs = np.array(z)  
ax.plot(xs, zs, x, zdir='x', color='b', marker='o', alpha=0.8, label = 'A装填方式')  
  
  
for i in range(0, 1):  
 z = sheet\_.col\_values(i)  
 x = x2\_[i]  
 xs = np.array(x1)  
 zs = np.array(z)  
 ax.plot(xs, zs, x, zdir='x', color='r', marker='o', alpha=0.8)  
  
z = sheet\_.col\_values(1)  
x = x2\_[1]  
xs = np.array(x1)  
zs = np.array(z)  
ax.plot(xs, zs, x, zdir='x', color='r', marker='o', alpha=0.8, label = 'B装填方式')  
  
plt.title('乙醇浓度与C4烯烃收率关系图', fontproperties = font)  
ax.set\_xlabel('乙醇浓度', fontproperties = font)  
ax.set\_zlabel('C4烯烃收率', fontproperties = font)  
ax.set\_ylabel('温度', fontproperties = font)  
plt.legend(loc = 'upper left', prop = font)  
#ax.legend(loc = 'upper left')  
plt.savefig("浓度1.png")  
plt.show()

④绝对质量与乙醇转化率关系图

#A1 A2 A4 A6  
import xlrd  
import random  
import pylab as mpl  
import matplotlib.pyplot as plt  
import numpy as np  
from matplotlib.font\_manager import FontProperties # 字体管理器  
  
# 设置汉字格式  
font = FontProperties(fname=r"consnerdi.ttf", size=12)  
workbook = xlrd.open\_workbook(r'绝对质量&转化率.xlsx')  
sheet = workbook.sheet\_by\_index(0)  
x1 = [10, 25, 50, 75, 100]  
x2 = [250, 275, 300, 350, 400]  
fig = plt.figure()  
ax = fig.add\_subplot(projection = '3d')  
  
for i in range(0, 5):  
 z = sheet.col\_values(i)  
 x = x1[i]  
 xs = np.array(x2)  
 zs = np.array(z)  
 ax.plot(xs, zs, x, zdir='x', color='r', marker='o', alpha=0.8)  
  
for i in range(0, 5):  
 z = sheet.row\_values(i)  
 x = x2[i]  
 xs = np.array(x1)  
 zs = np.array(z)  
 ax.plot(xs, zs, x, zdir='y', color='r', marker='o', alpha=0.8)  
  
plt.title('绝对质量与乙醇转化率关系图', fontproperties = font)  
ax.set\_xlabel('绝对质量', fontproperties = font)  
ax.set\_zlabel('乙醇转化率', fontproperties = font)  
ax.set\_ylabel('温度', fontproperties = font)  
#ax.legend(loc = 'upper left')  
plt.savefig("绝对质量&转化率.png")  
plt.show()

绝对质量与C4烯烃选择性关系图

#A1 A2 A4 A6  
import xlrd  
import random  
import pylab as mpl  
import matplotlib.pyplot as plt  
import numpy as np  
from matplotlib.font\_manager import FontProperties # 字体管理器  
  
# 设置汉字格式  
font = FontProperties(fname=r"consnerdi.ttf", size=12)  
workbook = xlrd.open\_workbook(r'绝对质量&选择性.xlsx')  
sheet = workbook.sheet\_by\_index(0)  
x1 = [10, 25, 50, 75, 100]  
x2 = [250, 275, 300, 350, 400]  
fig = plt.figure()  
ax = fig.add\_subplot(projection = '3d')  
  
for i in range(0, 5):  
 z = sheet.col\_values(i)  
 x = x1[i]  
 xs = np.array(x2)  
 zs = np.array(z)  
 ax.plot(xs, zs, x, zdir='x', color='r', marker='o', alpha=0.8)  
  
for i in range(0, 5):  
 z = sheet.row\_values(i)  
 x = x2[i]  
 xs = np.array(x1)  
 zs = np.array(z)  
 ax.plot(xs, zs, x, zdir='y', color='r', marker='o', alpha=0.8)  
  
plt.title('绝对质量与C4烯烃选择性关系图', fontproperties = font)  
ax.set\_xlabel('绝对质量', fontproperties = font)  
ax.set\_zlabel('C4烯烃选择性', fontproperties = font)  
ax.set\_ylabel('温度', fontproperties = font)  
#ax.legend(loc = 'upper left')  
plt.savefig("绝对质量&选择性.png")  
plt.show()

⑤装填方式与乙醇转化率关系图

#A1 A2 A4 A6  
import xlrd  
import random  
import pylab as mpl  
import matplotlib.pyplot as plt  
import numpy as np  
from matplotlib.font\_manager import FontProperties # 字体管理器  
  
# 设置汉字格式  
font = FontProperties(fname=r"consnerdi.ttf", size=12)  
workbook = xlrd.open\_workbook(r'AB&转化率.xlsx')  
sheet = workbook.sheet\_by\_index(0)  
x1 = [1.68, 2.1]  
x2 = [250, 275, 300, 350, 400]  
fig = plt.figure()  
ax = fig.add\_subplot(projection = '3d')  
  
for i in range(0, 2):  
 z = sheet.col\_values(i)  
 x = x1[i]  
 xs = np.array(x2)  
 zs = np.array(z)  
 if i == 0:  
 ax.plot(xs, zs, x, zdir='x', color='b', marker='o', alpha=0.8)  
 else:  
 l1 = ax.plot(xs, zs, x, zdir='x', color='b', marker='o', alpha=0.8, label = 'A装填')  
  
for i in range(2, 4):  
 z = sheet.col\_values(i)  
 x = x1[i - 2]  
 xs = np.array(x2)  
 zs = np.array(z)  
 if i == 2:  
 ax.plot(xs, zs, x, zdir='x', color='r', marker='o', alpha=0.8)  
 else:  
 l2 = ax.plot(xs, zs, x, zdir='x', color='r', marker='o', alpha=0.8, label='B装填')  
  
plt.title('装填方式与乙醇转化率关系图', fontproperties = font)  
ax.set\_xlabel('乙醇浓度', fontproperties = font)  
ax.set\_zlabel('乙醇转化率', fontproperties = font)  
ax.set\_ylabel('温度', fontproperties = font)  
plt.legend(loc = 'upper left', prop = font)  
#ax.legend(loc = 'upper left')  
plt.savefig("AB&转化率.png")  
plt.show()

神经网络

function value = func(x)  
 value = func1(x) \* func2(x) / 100;  
end

function value = func1(x)  
 load('func1', 'net');  
 load('func1', 'inputps');  
 load('func1', 'outputps');  
   
 an = sim(net, mapminmax('apply',x,inputps));  
 value = mapminmax('reverse',an,outputps);  
end

function value = func2(x)  
 load('func2', 'net');  
 load('func2', 'inputps');  
 load('func2', 'outputps');  
   
 an = sim(net, mapminmax('apply',x,inputps));  
 value = mapminmax('reverse',an,outputps);  
end

clc  
clear  
close all  
[input\_data] = textread('input.txt') ;  
[output\_data] = textread('output1.txt');  
input = input\_data;  
output = output\_data;  
  
c = randperm(74);  
A = input, B = output;  
for i = 1:4  
 A(c(i)) = input(i);  
 B(c(i)) = output(i);  
end  
input = A, output = B;  
  
input\_train = input(1:74,:)';  
output\_train = output(1:74,:)';  
input\_test = input(1:20,:)';  
output\_test = output(1:20,:)';  
  
%训练数据归一化  
[inputn, inputps] = mapminmax(input\_train);  
[outputn, outputps] = mapminmax(output\_train);  
net = newff(inputn,outputn,90);  
  
%参数设置  
net.trainParam.epochs=1000; %迭代次数  
net.trainParam.lr=0.3; %学习率  
net.trainParam.goal=0.0000000001; %收敛目标  
  
%神经网络训练  
net = train(net, inputn, outputn);  
load('func1', 'net');  
save('func1', 'net', 'inputps', 'outputps');  
%训练数据归一化  
inputn\_test = mapminmax('apply', input\_test, inputps);  
%神经网络测试输出  
an = sim(net, inputn\_test);  
pred\_output = mapminmax('reverse',an,outputps);  
  
%可视化处理  
figure(1)  
plot(pred\_output(1,:),'r')  
hold on  
plot(output\_test(1,:),'b.');  
legend('模拟值（乙醇转化率）','原始值（乙醇转化率）')  
err = abs(pred\_output - output\_test);  
err\_mean = mean(err);  
title('原始值与模拟值')  
xlabel('样本')  
ylabel('乙醇转化率')  
  
figure(2)  
plot(err,'-\*')  
err\_mean  
title('测试误差')  
ylabel('平均误差')  
xlabel('样本')

clc  
clear  
close all  
[input\_data] = textread('input.txt') ;  
[output\_data] = textread('output2.txt');  
input = input\_data;  
output = output\_data;  
  
c = randperm(74);  
A = input, B = output;  
for i = 1:4  
 A(c(i)) = input(i);  
 B(c(i)) = output(i);  
end  
input = A, output = B;  
  
input\_train = input(1:74,:)';  
output\_train = output(1:74,:)';  
input\_test = input(1:20,:)';  
output\_test = output(1:20,:)';  
  
%训练数据归一化  
[inputn, inputps] = mapminmax(input\_train);  
[outputn, outputps] = mapminmax(output\_train);  
net = newff(inputn,outputn,90);  
  
%参数设置  
net.trainParam.epochs=1000; %迭代次数  
net.trainParam.lr=0.3; %学习率  
net.trainParam.goal=0.0000000001; %收敛目标  
  
%神经网络训练  
net = train(net, inputn, outputn);  
load('func2', 'net');  
save('func2', 'net', 'inputps', 'outputps');  
%训练数据归一化  
inputn\_test = mapminmax('apply', input\_test, inputps);  
%神经网络测试输出  
an = sim(net, inputn\_test);  
pred\_output = mapminmax('reverse',an,outputps);  
  
%可视化处理  
figure(1)  
plot(pred\_output(1,:),'r')  
hold on  
plot(output\_test(1,:),'b.');  
legend('模拟值（C4烯烃选择性）','原始值（C4烯烃选择性）')  
err = abs(pred\_output - output\_test);  
err\_mean = mean(err);  
title('原始值与模拟值')  
xlabel('样本')  
ylabel('C4烯烃选择性')  
  
figure(2)  
plot(err,'-\*')  
err\_mean  
title('测试误差')  
ylabel('平均误差')  
xlabel('样本')

问题三：

模拟退火

T0=99588; % 初始温度  
T1=1e-3;% 终止温度  
l=2; % 各温度下的迭代次数  
q=0.993;%降温速率  
Time = ceil(log(T1 / T0) / log(0.9)); %提前估计迭代次数  
point1=[150; 1; 2.5; 1.68; 325];  
f = zeros(Time,1);%存储退火过程中的函数值  
f0=func(point1);%计算初始值  
idc=0;%初始化计数值  
   
while T0>T1 && idc <= 100  
 idc =idc+1;  
 point2=new\_point(point1, T0);  
 % Metropolis法则判断是否接受新解  
 [point1,R] = Metropolis(point1,point2,T0);  
 if idc == 1 || R > f(idc-1)  
 f(idc) = R;   
 else  
 f(idc) = f(idc-1);%如果当前温度下函数值大于上一路程则记录上一函数值  
 end  
 T0 = q \* T0;   
end  
  
figure  
x = 1:idc;  
plot(x,f(x))  
xlabel('迭代次数')  
ylabel('C4烯烃收率（%）')  
title('无限制下的模拟退火')  
   
disp('最优解:')  
f(idc)

function value = func(x)  
 value = func1(x) \* func2(x) / 100;  
 value = max(value, unifrnd(0, 6));  
 value = min(value, unifrnd(41, 48.24586));  
end

function value = func1(x)  
 load('func1', 'net');  
 load('func1', 'inputps');  
 load('func1', 'outputps');  
   
 an = sim(net, mapminmax('apply',x,inputps));  
 value = mapminmax('reverse',an,outputps);  
end

function value = func2(x)  
 load('func2', 'net');  
 load('func2', 'inputps');  
 load('func2', 'outputps');  
   
 an = sim(net, mapminmax('apply',x,inputps));  
 value = mapminmax('reverse',an,outputps);  
end

function [S,R] = Metropolis(S1,S2,T)  
   
% S1： 当前解  
% S2: 新解  
% D: 距离矩阵（点的函数值）  
% T: 当前温度  
% S： 下一个当前解  
% R： 下一个当前解的函数值  
  
R1 = func(S1);   
R2 = func(S2);   
dC = R2 - R1; %计算函数值之差  
if dC > 0 %如果函数值增加 接受新点  
 S = S2;  
 R = R2;  
elseif exp(dC/T)>= rand %以exp(-dC/T)概率接受新点  
 S = S2;  
 R = R2;  
else   
 S = S1;  
 R = R1;  
end

function newpoint = new\_point(last\_point, T)  
 f = unifrnd(-1, 1);  
 newpoint(1,1) = last\_point(1, 1) + unifrnd(100, 400) \* T \* 0.00005 \* f;  
 newpoint(2,1) = last\_point(2, 1) + unifrnd(0, 2) \* T \* 0.00005 \* f;  
 newpoint(3,1) = last\_point(3, 1) + unifrnd(0.5, 5) \* T \* 0.00005 \* f;  
 newpoint(4,1) = last\_point(4, 1) + unifrnd(0.3, 2.1) \* T \* 0.00005 \* f;  
 newpoint(5,1) = last\_point(5, 1) + unifrnd(250, 400) \* T \* 0.00005 \* f;  
end