

25. 人工神经网络的 1stOpt 代码级实现及应用

25.1 背景

人工智能无疑是当下科技行业热点，而深度学习又是人工智能的基石。深度学习是以人工神经网络为基础的，理解并掌握人工神经网络的基本原理并加以实现和应用，无疑对人工智能基本原理和框架的了解是有帮助的。

人工神经网络模型从原理上可以简单地归结为一个优化求解或者多元非线性拟合问题，其所谓的学习训练过程实际上就是一个优化迭代求参过程，虽然 1stOpt 附带有神经网络工具箱，非常方便使用，但基于 1stOpt 平台，从代码级实现简单的人工神经网络并加以应用，对普通用户加深了解还是有相当益处的。

25.2 人工神经网络

典型的三层神经网络如图 25-1 示，一般包括一个输入层，一个隐含层和一个输出层，相邻层节点之间的连线称之为连接权重值，实际就是一个待定参数；节点转换函数有多种，常用 Sigmoid 和 Tanh 两种，如下公式。

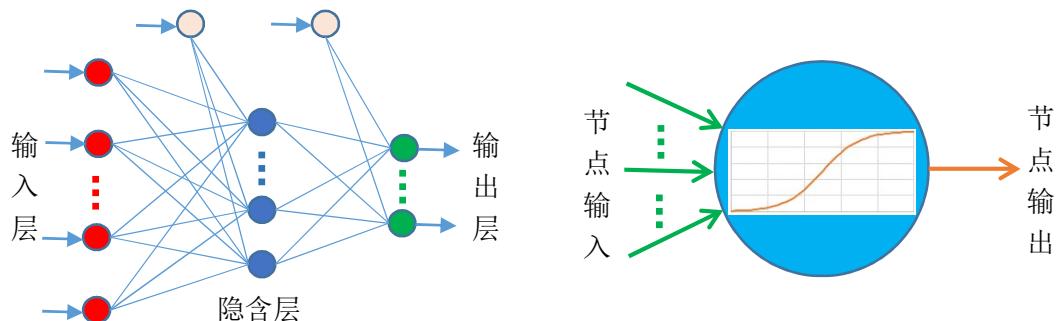
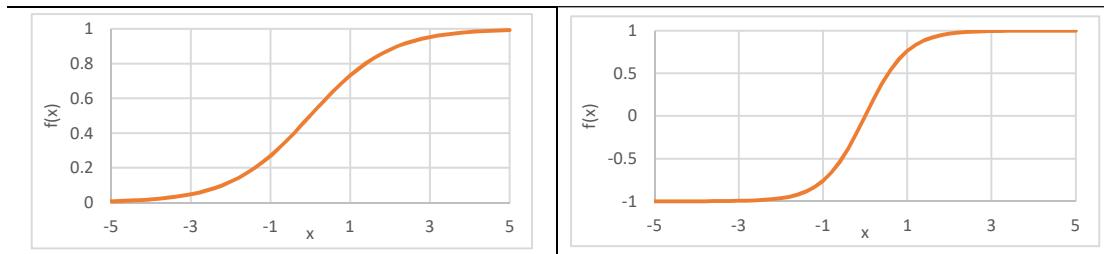


图 25-1 典型三层神经网络

表 25.1 转换函数

Sigmoid 函数	Tanh 函数
$f(x) = \frac{1}{1 + \exp(-x)}$	$f(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$



25.3 研究案例

选取四个案例，一个是著名的 XOR（异或）问题，另外三个是数据拟合问题，拟合问题 1 是单自变量（输入），拟合问题 2 是双自变量（输入），拟合问题 3 是峰函数拟合。

表 25.2 XOR 案例数据

X1 (输入)	X2 (输入)	Y (输出)
0	0	0
0	1	1
1	0	1
1	1	0

表 25.3 拟合案例数据之一

自变量 X	0.05,0.15,0.25,0.35,0.45,0.55,0.65,0.75,0.85,0.95,1.05,1.15,1.25,1.35,1.45;
因变量 Y	0.13,0.13,0.19,0.34,0.53,0.71,1.06,1.6,1.64,1.83,2.09,2.05,2.13,2.12,2.09;

表 25.4 拟合案例数据之二

自变量 X1	0,0.6,0.6,1.4,2.6,3.2,0.8,1.6,2.6,4.0,1.2,2.0,0.4,6,3.2,1.6,4.2,2.0,3.2,2.8,4.2,5.4,5.6,3.2
自变量 X2	0,0.4,1.0,1.4,1.4,1.6,2.0,2.2,2.2,2.2,2.6,2.6,2.8,3.0,3.2,3.4,3.8,3.8,4.2,4.2,4.4,4.8,5.0
因变量 Y	40.2,11.0349,4.48869,2.46137,2.45137,1.82343,1.00094,0.741352,0.741352, 0.741352,0.406863,0.406862,0.301411,0.223291,0.165418,0.122545,0.067254, 0.067254,0.036910,0.036910,0.027343,0.015006,0.011117

表 25.5 拟合案例数据之三

自变量 X	1,7,13,19,25,31,37,43,49,55,61,67,73,79,85,91,97,103,109,115,121,127,133,139,145,151,157,163, 169,175,181,187,193,199,205,211,217,223,229,235,241,247
因变量 Y	97.59,90.21,87.13,81.29,77.17,66.92,69.72,59.14,57.13,57.43,52.32,53.29,60.15,67.77,80.13, 93.47,119.62,129.72,133.83,122.03,103.45,92.50,76.13,79.74,87.17,93.69,88.12,72.66,52.80, 39.28,20.27,17.26,12.33,11.76,14.27,9.73,9.02,10.23,12.45,8.39,10.32,8.98

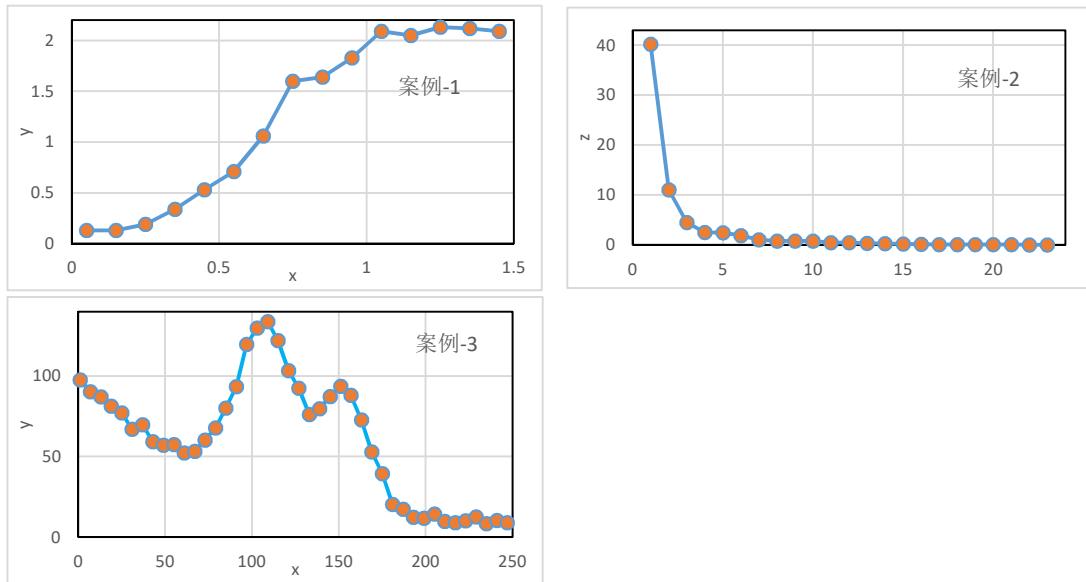


图 25-2 拟合案例数据散点图

25.4 神经网络代码实现

进行神经网络训练计算时一般要对输入及输出数据先进行归一化处理，一般而言，对输入实际数据，可归一至[0,1]之间，采用公式（25-1）；而对输出实际数据，则要根据输出层采用的转换函数，比如，如果输出层采用 Sigmoid 函数，则采用公式（25-3），可归一至[0.1,0.9]之间，tanh 函数，采用公式（25-5）归一至[-0.9,0.9]之间。输出实际数据归一后的上下限值最好要略小于输出层转换函数的上下值。对应的还原公式分别见公式（25-2）、（25-4）和（25-6）。

表 25.6 数据均以及还原公式

归一范围	用途	公式
[0,1]	归一	$D_{new} = \frac{D_{old}-D_{min}}{D_{max}-D_{min}}$ (25-1)
	还原	$D_{old} = D_{min} + D_{new} \cdot (D_{max} - D_{min})$ (25-2)
[0.1,0.9]	归一	$D_{new} = 0.1 + 0.8 \cdot \frac{D_{old}-D_{min}}{D_{max}-D_{min}}$ (25-3)
	还原	$D_{old} = D_{min} + \frac{(D_{new}-0.1) \cdot (D_{max}-D_{min})}{0.8}$ (25-4)
[-0.9,0.9]	归一	$D_{new} = 0.9 \cdot \left(2 \cdot \frac{D_{old}-D_{min}}{D_{max}-D_{min}} - 1 \right)$ (25-5)
	还原	$D_{old} = D_{min} + \frac{(D_{new}-0.1) \cdot (D_{max}-D_{min})}{0.9}$ (25-6)

其中： D_{old} 、 D_{min} 、 D_{max} 和 D_{new} 分别代表原数据，原数据最小，原数据最大。

25.4.1 XOR 问题

XOR 问题也称异或问题，是对两个运算元的一种逻辑分析类型，当两两数值相同时为否（0），而数值不同时为真（1）。该问题有两个输入一个输出，一个中间隐含层，隐含层节点数取 2，可称之为“2-2-1”神经网络结构，有 P1 至 P9 共 9 个连接权重，也即 9 个待求参数，分别在快捷模式及编程模式（Fortran 及 Pascal）代码实现，注意 Fortran 代码中“Sigmoid”不是缺省的内置函数，因此自定义该函数。

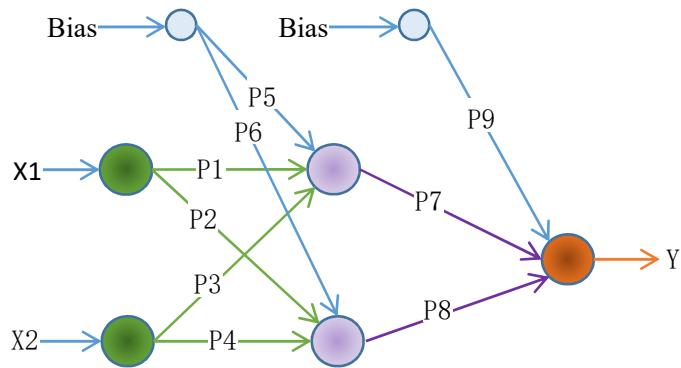


图 25-3 XOR 问题神经网络图

代码 25.1 XOR 问题求解代码：快捷模式

```

Constant xmin=0,xmax=1,ymin=0,ymax=1;
ConstStr s1 = x1*p1+x2*p3+p5,      //隐含层第一个节点输入
           s2 = x1*p2+x2*p4+p6,      //隐含层第二个节点输入
           h1 = Sigmoid(s1),          //隐含层第一个节点转换输出值
           h2 = Sigmoid(s2),          //隐含层第二个节点转换输出值
           m1 = h1*p7+h2*p8+p9;     //输出层节点输入

Parameter p(9);
Variable x1,x2, y=[0.1+0.8*(y-ymin)/(ymax-ymin)]; //实际输出值均一化至[0.1,0.9]
Function y = Sigmoid(m1);
Data;
//X1 X2 Y
0    0    0
0    1    1
1    0    1
1    1    0
  
```

上述代码中的模型公式实际如下：

$$y = \frac{1}{1 + \exp\left(-\left(\frac{p_7}{1 + \exp(-(x_1 \cdot p_1 + x_2 \cdot p_3 + p_5))} + \frac{p_8}{1 + \exp(-(x_1 \cdot p_2 + x_2 \cdot p_4 + p_6))} + p_9\right)\right)}$$
(25-7)

代码 25.2 XOR 问题求解代码：Fortran 编程格式

```

Constant xmin=0,xmax=1,ymin=0,ymax=1;
ConstStr Sigmoid(v)=1/(1+exp(-v));
Parameter p(9);
Variable x1,x2, y=[0.1+0.8*(y-ymin)/(ymax-ymin)]; //实际输出值均一化至[0.1,0.9]
StartProgram [Fortran];
Subroutine MainModel
    integer i
    real*8 s1, s2, h1, h2, m1
    do i = 0, DataLength - 1
        s1 = x1(i)*p1+x2(i)*p3+p5 !隐含层第一个节点输入
        s2 = x1(i)*p2+x2(i)*p4+p6 !隐含层第二个节点输入
        h1 = Sigmoid(s1)           !隐含层第一个节点转换输出值
        h2 = Sigmoid(s2)           !隐含层第二个节点转换输出值
        m1 = h1*p7+h2*p8+p9      !输出层节点输入
        y(i) = Sigmoid(m1)         !输出层节点输出
    end do
End Subroutine
EndProgram;
Data;
//X1 X2 Y
0   0   0
0   1   1
1   0   1
1   1   0

```

代码 25.3 XOR 问题求解代码：Pascal 编程格式

```

Constant xmin=0,xmax=1,ymin=0,ymax=1;
Parameter p(9);
Variable x1,x2, y=[0.1+0.8*(y-ymin)/(ymax-ymin)]; //实际输出值均一化至[0.1,0.9]
StartProgram [Pascal];
Procedure MainModel;
var i: integer;
    s1, s2, h1, h2, m1: double;
Begin
    for i := 0 to DataLength - 1 do begin
        s1 := x1[i]*p1+x2[i]*p3+p5; //隐含层第一个节点输入
        s2 := x1[i]*p2+x2[i]*p4+p6; //隐含层第二个节点输入
        h1 := Sigmoid(s1);          //隐含层第一个节点转换输出值
        h2 := Sigmoid(s2);          //隐含层第二个节点转换输出值
        m1 := h1*p7+h2*p8+p9;      //输出层节点输入
        y[i] := Sigmoid(m1);         //输出层节点输出
    end;
End;
EndProgram;
Data;
//X1 X2 Y
0   0   0
0   1   1
1   0   1
1   1   0

```

结果之一：

```

Sum Squared Error (SSE): 0
Root of Mean Square Error (RMSE): 0
Correlation Coef. (R): 1
R-Square: 1
Adjusted R-Square: -100
Determination Coef. (DC): 1

```

Parameter	Best Estimate
p1	542.320662412911
p2	549.846628662437
p3	-407.990943360265
p4	-114.345834896753
p5	369.775978768972
p6	-503.644127033774
p7	-1015.07110687369
p8	871.337709638193
p9	611.846880436393

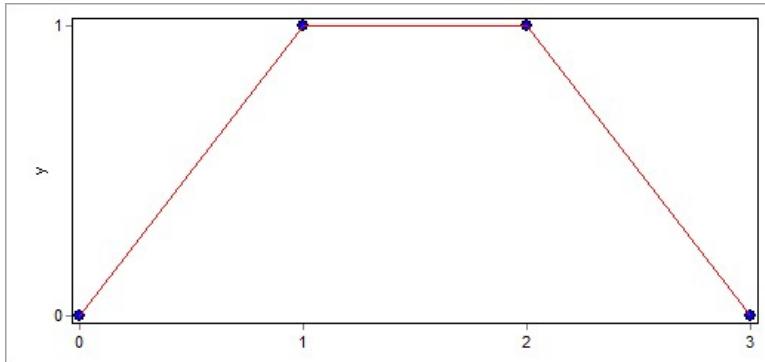


图 25-4 XOR 问题计算结果图

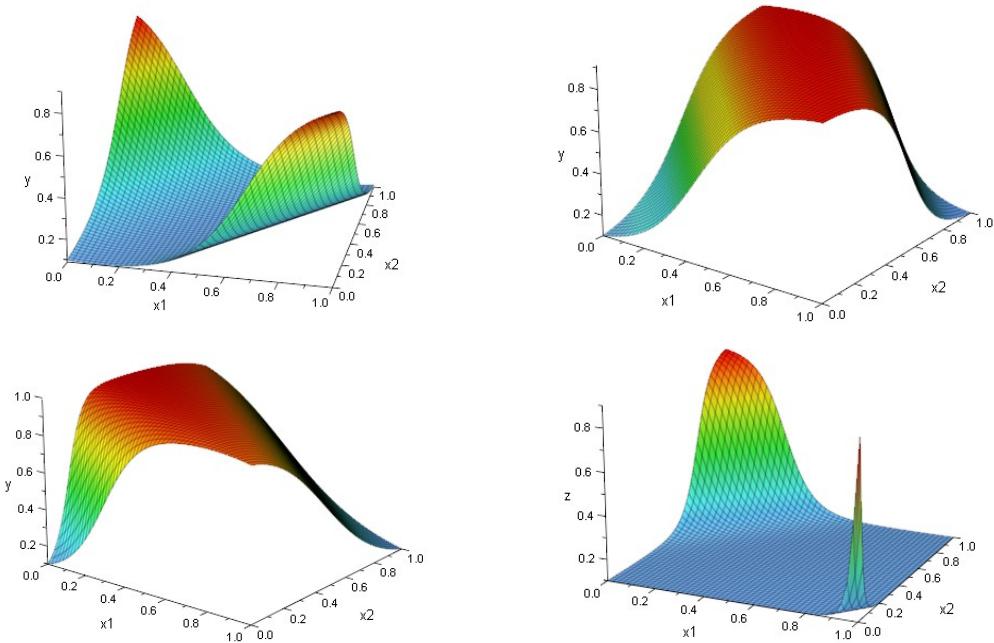


图 25-5 XOR 问题计算结果三维图

从吻合度看计算结果是非常完美的，只是参数值是多解的，这一现象有称之为神经网络特性的，也有视作缺点的，从拟合角度看是过拟合的。

如果隐含层节点改为一个，连接方式如图 25-6 示，可称之为“2-1-1”神经网络结构，该神经网络应该是求解 XOR 问题最简洁形式了，对应求解代码仅给出快捷模式。

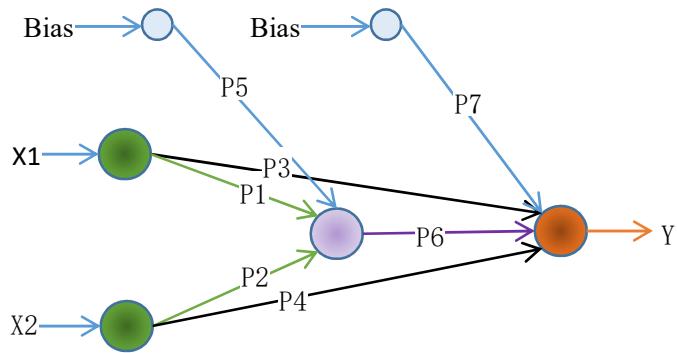


图 25-6 XOR 问题单隐含层节点神经网络图

代码 25.4 XOR 问题求解代码：快捷模式

```

Constant xmin=0,xmax=1,ymin=0,ymax=1;
ConstStr s1 = x1*p1+x2*p2+p5,           //隐含层第一个节点输入
          h1 = Sigmoid(s1),             //隐含层第一个节点转换输出值
          m1 = h1*p6+x1*p3+x2*p4+p7;   //输出层节点输入

Parameter p(7);
Variable x1,x2, y=[0.1+0.8*(y-ymin)/(ymax-ymin)]; //实际输出值均一化至[0.1,0.9]
Function y = Sigmoid(m1);

Data;
//X1 X2 Y
0    0    0
0    1    1
1    0    1
1    1    0

```

对应模型公式实际如下：

$$y = \frac{1}{1 + \exp\left(-\left(\frac{p_6}{1 + \exp(-(x_1 \cdot p_1 + x_2 \cdot p_2 + p_5))} + x_1 \cdot p_3 + x_2 \cdot p_4 + p_7\right)\right)} \quad (25-8)$$

25.4.2 拟合问题 1-单自变量输入

该问题是一个自变量一个因变量的拟合问题，分别采用两种神经网络结构，即 3 个隐含层节点的“1-3-1”结构及 2 个隐含层节点的“1-2-1”结构，见下图。

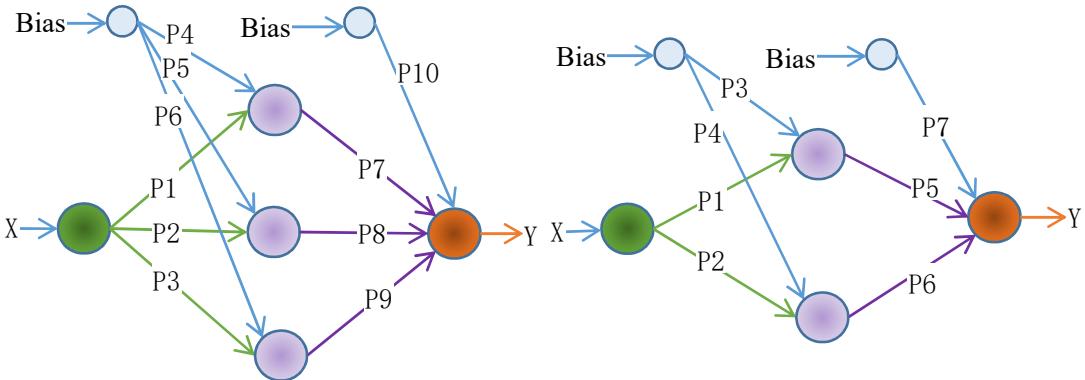


图 25-7 拟合问题 1 神经网络“1-3-1”结构 图 25-8 拟合问题 1 神经网络“1-2-1”结构

代码 25.5 拟合问题 1 求解代码：1-3-1 神经网络结构

```

Constant xmin=0.05,xmax=1.45,ymin=0.13,ymax=2.13;
ConstStr s1 = x*p1+p4,           //隐含层第一个节点输入
            s2 = x*p2+p5,           //隐含层第二个节点输入
            s3 = x*p3+p6,           //隐含层第三个节点输入
            h1 = Sigmoid(s1),        //隐含层第一个节点转换输出值
            h2 = Sigmoid(s2),        //隐含层第二个节点转换输出值
            h3 = Sigmoid(s3),        //隐含层第三个节点转换输出值
            m1 = h1*p7+h2*p8+h3*p9+p10; //输出层节点输入

Parameter p(10);

Variable x=[(x-xmin)/(xmax-xmin)], y=[0.1+0.8*(y-ymin)/(ymax-ymin)];
Function y = Sigmoid(m1);          //输出层节点输出

Data;
x=0.05,0.15,0.25,0.35,0.45,0.55,0.65,0.75,0.85,0.95,1.05,1.15,1.25,1.35,1.45;
y=0.13,0.13,0.19,0.34,0.53,0.71,1.06,1.6,1.64,1.83,2.09,2.05,2.13,2.12,2.09;

```

代码 25.6 拟合问题 1 求解代码：1-2-1 神经网络结构

```

Constant xmin=0.05,xmax=1.45,ymin=0.13,ymax=2.13;
ConstStr s1 = x*p1+p3,           //隐含层第一个节点输入
            s2 = x*p2+p4,           //隐含层第二个节点输入
            h1 = Sigmoid(s1),        //隐含层第一个节点转换输出值
            h2 = Sigmoid(s2),        //隐含层第二个节点转换输出值
            m1 = h1*p5+h2*p6+p7;    //输出层节点输入

Parameter p(7);

Variable x=[(x-xmin)/(xmax-xmin)], y=[0.1+0.8*(y-ymin)/(ymax-ymin)];
Function y = Sigmoid(m1);          //输出层节点输出

Data;
x=0.05,0.15,0.25,0.35,0.45,0.55,0.65,0.75,0.85,0.95,1.05,1.15,1.25,1.35,1.45;
y=0.13,0.13,0.19,0.34,0.53,0.71,1.06,1.6,1.64,1.83,2.09,2.05,2.13,2.12,2.09;

```

1-3-1 神经网络结果：

```

Sum Squared Error (SSE): 0.000674237702764238
Root of Mean Square Error (RMSE): 0.00670441497703435
Correlation Coef. ®: 0.999778170181158
R-Square: 0.999556389570785
Adjusted R-Square: 0.999482454499249
Determination Coef. (DC): 0.999556389565581
Chi-Square: 0.000571002192032982
F-Statistic: 1251.78903136308

```

Parameter	Best Estimate
-----------	---------------

p1	-15.93838024817
p2	58.7858444037697
p3	53.9210735178917
p4	3.68281173776651
p5	-38.4745951437337
p6	-23.7657298251657
p7	-1.77623216116366
p8	1.19414039734674
p9	1.3762942686991
p10	-0.508648350269947

1-2-1 神经网络结果:

Sum Squared Error (SSE): 0.00484418397327658
Root of Mean Square Error (RMSE): 0.0179706871938287
Correlation Coef. ®: 0.998405737645675
R-Square: 0.996814016963804
Adjusted R-Square: 0.996283019791104
Determination Coef. (DC): 0.996812799776722
Chi-Square: 0.006793819098701
F-Statistic: 416.586946114894

Parameter Best Estimate

-----	-----
p1	-501.70327047945
p2	7.82258184774099
p3	284.032462345087
p4	-3.58415802130978
p5	0.526847599533944
p6	5.12161932329402
p7	-2.79449793720496

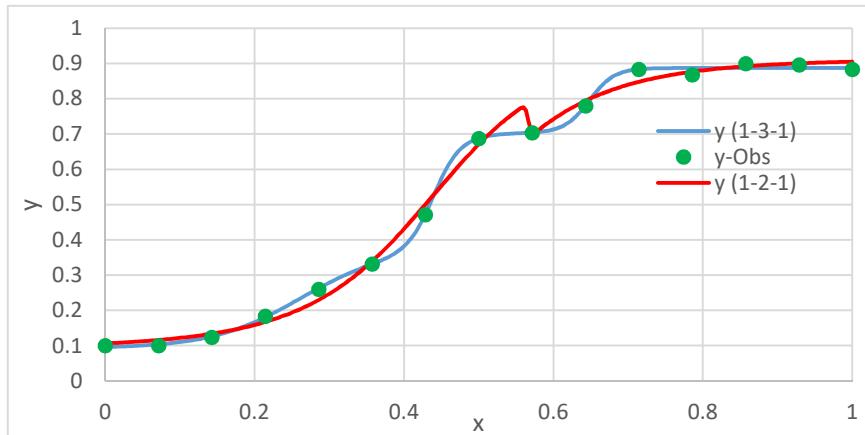


图 25-9 拟合问题 1 结果对比图

25.4.3 拟合问题 2-两自变量输入

该拟合问题是两输入（两个自变量）一输出（一个因变量），神经网络结构可采用前面 XOR 问题相同的“2-2-1”结构，参见图 25-3。

代码 25.7 拟合问题 2 求解代码

```

Constant xmin1=0,xmax1=5.6,xmin2=0,xmax2=5,ymin=0.011117,ymax=40.2;
Parameter p(9);
Variable x1=[(x1-xmin1)/(xmax1-xmin1)],x2=[(x2-xmin2)/(xmax2-xmin2)],y=[0.1+0.8*(y-ymin)/(ymax-
ymin)];
ConstStr s1 = x1*p1+x2*p3+p5,      //隐含层第一个节点输入
           s2 = x1*p2+x2*p4+p6,      //隐含层第二个节点输入
           h1 = Sigmoid(s1),         //隐含层第一个节点转换输出值
           h2 = Sigmoid(s2),         //隐含层第二个节点转换输出值
           m1 = h1*p7+h2*p8+p9;     //输出层节点输入
//Variable x1,x2,y;
Variable x1,x2, y=[0.1+0.8*(y-ymin)/(ymax-ymin)]; //实际输出值均一化至[0.1,0.9]
Function y = Sigmoid(m1);
Data;
x1=0,0.6,0.6,1.4,2.6,3.2,0.8,1.6,2.6,4.0,1.2,2.0,4.6,3.2,1.6,4.2,2.0,3.2,2.8,4.2,5.4,5.6,3.2;
x2=0,0.4,1.0,1.4,1.4,1.6,2.0,2.2,2.2,2.2,2.6,2.6,2.8,3.0,3.2,3.4,3.8,3.8,4.2,4.2,4.4,4.8,5.0;
y=40.2,11.0349,4.48869,2.46137,2.45137,1.82343,1.00094,0.741352,0.741352,0.741352,0.406863,0.406862,0.
301411,0.223291,0.165418,0.122545,0.067254,0.067254,0.036910,0.036910,0.027343,0.015006,0.011117;

```

结果：

```

Sum Squared Error (SSE): 9.12118632500485E-9
Root of Mean Square Error (RMSE): 1.99141487008225E-5
Correlation Coef. (R): 0.999999992706334
R-Square: 0.999999985412668
Adjusted R-Square: 0.999999983109405
Determination Coef. (DC): 0.999999985412668
Chi-Square: 3.36966246385656E-8
F-Statistic: 119967107.438314

```

Parameter	Best Estimate
p1	0.00409499258040609
p2	-12.6286469373196
p3	7.49437971360504
p4	-19.8804782141023
p5	-0.456012743501748
p6	-9.88681945456666
p7	-2.80587102953835
p8	52706.1899870588
p9	0.606245002471331

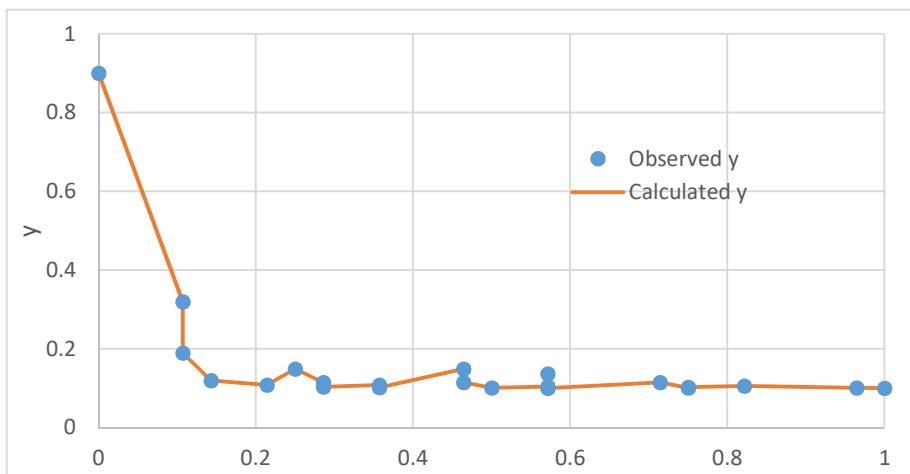


图 25-10 拟合问题 2 结果对比图

25.4.4 拟合问题 3-峰函数拟合

该峰值拟合问题采用下面四种神经网络结构：单隐含层“1-3-1”和“1-6-1”结构，两个隐含层“1-3-2-1”和“1-2-3-1”结构，分别见图 25-7，图 25-11，图 25-12 和图 25-13。

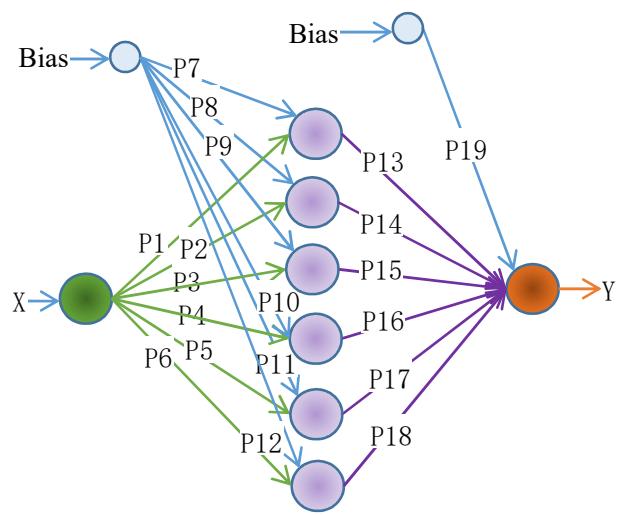


图 25-11 单隐含层“1-6-1”结构

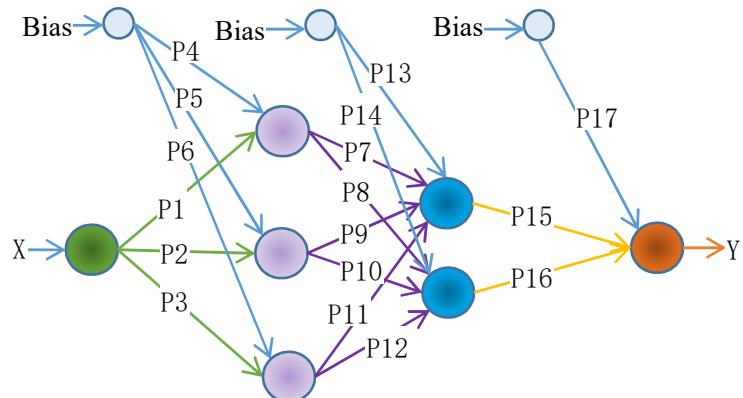


图 25-12 双隐含层“1-3-2-1”结构

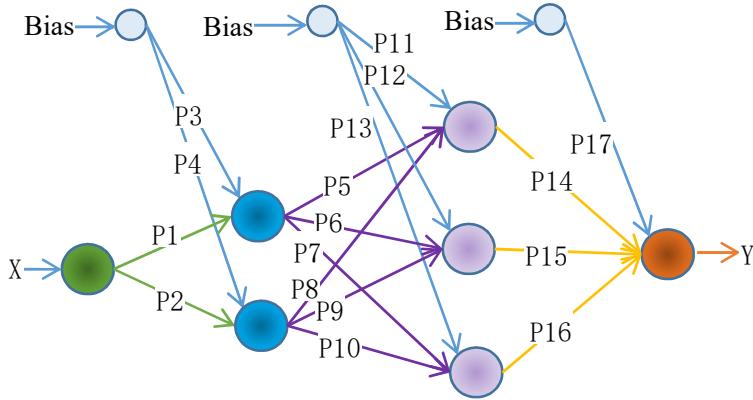


图 25-13 双隐含层“1-2-3-1”结构

代码 25.8 拟合问题 3 求解代码：“1-3-1”结构

```

Constant xmin=1,xmax=250,ymin=3.91,ymax=133.83;
ConstStr s1 = x*p1+p4,           //隐含层第一个节点输入
            s2 = x*p2+p5,           //隐含层第二个节点输入
            s3 = x*p3+p6,           //隐含层第三个节点输入
            h1 = Sigmoid(s1),        //隐含层第一个节点转换输出值
            h2 = Sigmoid(s2),        //隐含层第二个节点转换输出值
            h3 = Sigmoid(s3),        //隐含层第三个节点转换输出值
            m1 = h1*p7+h2*p8+h3*p9+p10; //输出层节点输入

Parameter p(10);
Variable x=[(x-xmin)/(xmax-xmin)], y=[0.1+0.8*(y-ymin)/(ymax-ymin)];
Function y = Sigmoid(m1);          //输出层节点输出
Data;
x=1,7,13,19,25,31,37,43,49,55,61,67,73,79,85,91,97,103,109,115,121,127,133,139,145,151,157,163,169,175,1
81,187,193,199,205,211,217,223,229,235,241,247;
y=97.59,90.21,87.13,81.29,77.17,66.92,69.72,59.14,57.13,57.43,52.32,53.29,60.15,67.77,80.13,93.47,119.62,1
29.72,133.83,122.03,103.45,92.50,76.13,79.74,87.17,93.69,88.12,72.66,52.80,39.28,20.27,17.26,12.33,11.76,1
4.27,9.73,9.02,10.23,12.45,8.39,10.32,8.98;

```

结果：

```

Sum Squared Error (SSE): 0.0224594425046305
Root of Mean Square Error (RMSE): 0.0231246325605972
Correlation Coef. (R): 0.994882678770112
R-Square: 0.989791544516795
Adjusted R-Square: 0.989268033979195
Determination Coef. (DC): 0.989762547842014
Chi-Square: 0.0331516859918789
F-Statistic: 344.443275476989

```

Parameter	Best Estimate
p1	14.0584561319597
p2	13.5632070002091
p3	40.182708188526
p4	-5.73974286378046
p5	-5.60359835699388
p6	-22.604411915481
p7	306.661203256644
p8	-315.074580557733
p9	5.61527627501533
p10	0.70900457070093

代码 25.9 拟合问题 3 求解代码：“1-6-1”结构

```
Constant xmin=1,xmax=250,ymin=3.91,ymax=133.83;
ConstStr s1 = x*p1+p7,           //隐含层第一个节点输入
           s2 = x*p2+p8,           //隐含层第二个节点输入
           s3 = x*p3+p9,           //隐含层第三个节点输入
           s4 = x*p4+p10,          //隐含层第四个节点输入
           s5 = x*p5+p11,          //隐含层第五个节点输入
           s6 = x*p6+p12,          //隐含层第六个节点输入
           h1 = Sigmoid(s1),        //隐含层第一个节点转换输出值
           h2 = Sigmoid(s2),        //隐含层第二个节点转换输出值
           h3 = Sigmoid(s3),        //隐含层第三节点转换输出值
           h4 = Sigmoid(s4),        //隐含层第四个节点转换输出值
           h5 = Sigmoid(s5),        //隐含层第五个节点转换输出值
           h6 = Sigmoid(s6),        //隐含层第六节点转换输出值
           m1 = h1*p13+h2*p14+h3*p15+h4*p16+h5*p17+h6*p18+p19; //输出层节点输入

Parameter p(19);
Variable x=[(x-xmin)/(xmax-xmin)], y=[0.1+0.8*(y-ymin)/(ymax-ymin)];
Function y = Sigmoid(m1);           //输出层节点输出
Data;
x=1,7,13,19,25,31,37,43,49,55,61,67,73,79,85,91,97,103,109,115,121,127,133,139,145,151,157,163,169,175,1
81,187,193,199,205,211,217,223,229,235,241,247;
y=97.59,90.21,87.13,81.29,77.17,66.92,69.72,59.14,57.13,57.43,52.32,53.29,60.15,67.77,80.13,93.47,119.62,1
29.72,133.83,122.03,103.45,92.50,76.13,79.74,87.17,93.69,88.12,72.66,52.80,39.28,20.27,17.26,12.33,11.76,1
4.27,9.73,9.02,10.23,12.45,8.39,10.32,8.98;
```

结果：

```
Sum Squared Error (SSE): 0.00463128532855455
Root of Mean Square Error (RMSE): 0.0105008903574372
Correlation Coef. (R): 0.998943939984406
R-Square: 0.997888995231569
Adjusted R-Square: 0.997780738576778
Determination Coef. (DC): 0.997888969774237
Chi-Square: 0.00753042352761268
F-Statistic: 604.187803348134
```

Parameter	Best Estimate
p1	-36.0706679697226
p2	-71.1794879656049
p3	-2233.67033673172
p4	29.0780922289408
p5	-10.286452303802
p6	-32.381246233618
p7	16.4849487900632
p8	36.8208442737502
p9	645.267577359264
p10	-19.542364299485
p11	0.51159222082519
p12	14.8165994921835
p13	75.1243832165293
p14	3.10575077982463
p15	-0.221410500364858
p16	-3.11787187886747
p17	2.27229879862323
p18	-79.9949636200053
p19	1.28345686490249

代码 25.10 拟合问题 3 求解代码：“1-3-2-1”结构

```
Constant xmin=1,xmax=250,ymin=3.91,ymax=133.83;
ConstStr s1 = x*p1+p4,           //第一隐含层第一个节点输入
          s2 = x*p2+p5,           //第一隐含层第二个节点输入
          s3 = x*p3+p6,           //第一隐含层第三个节点输入
          h1 = Sigmoid(s1),        //第一隐含层第一个节点转换输出值
          h2 = Sigmoid(s2),        //第一隐含层第二个节点转换输出值
          h3 = Sigmoid(s3),        //第一隐含层第三节点转换输出值
          v1 = h1*p7+h2*p9+h3*p11+p13, //第二隐含层第一个节点输入
          v2 = h1*p8+h2*p10+h3*p12+p14, //第二隐含层第二个节点输入
          f1 = Sigmoid(v1),         //第二隐含层第一个节点转换输出值
          f2 = Sigmoid(v2),         //第二隐含层第二个节点转换输出值
          m1 = f1*p15+f2*p16+p17;  //输出层节点输入

Parameter p(17);
Variable x=[(x-xmin)/(xmax-xmin)], y=[0.1+0.8*(y-ymin)/(ymax-ymin)];
Function y = Sigmoid(m1);           //输出层节点输出
Data;
x=1,7,13,19,25,31,37,43,49,55,61,67,73,79,85,91,97,103,109,115,121,127,133,139,145,151,157,163,169,175,1
81,187,193,199,205,211,217,223,229,235,241,247;
y=97.59,90.21,87.13,81.29,77.17,66.92,69.72,59.14,57.13,57.43,52.32,53.29,60.15,67.77,80.13,93.47,119.62,1
29.72,133.83,122.03,103.45,92.50,76.13,79.74,87.17,93.69,88.12,72.66,52.80,39.28,20.27,17.26,12.33,11.76,1
4.27,9.73,9.02,10.23,12.45,8.39,10.32,8.98;
```

结果：

```
Sum Squared Error (SSE): 0.00588025463313667
Root of Mean Square Error (RMSE): 0.0118324157589957
Correlation Coef. (R): 0.998659002203419
R-Square: 0.997319802681928
Adjusted R-Square: 0.997182356665617
Determination Coef. (DC): 0.997319665193332
Chi-Square: 0.00878103741188284
F-Statistic: 580.967658728095
```

Parameter	Best Estimate
p1	-10.7362088924202
p2	108.777102388098
p3	-12.9070821032782
p4	4.62918057329104
p5	-37.4246881340047
p6	5.37072915797885
p7	20.5863637277871
p8	-92.825388833024
p9	87.1941864574144
p10	1.06368988351193
p11	-21.7117950518948
p12	91.3332053864856
p13	-94.9442152118019
p14	2.0778524328004
p15	-3032.2184943356
p16	-5.62385689103931
p17	4.87191272579562

代码 25.11 拟合问题 3 求解代码：“1-2-3-1”结构

```
Constant xmin=1,xmax=250,ymin=3.91,ymax=133.83;
ConstStr s1 = x*p1+p3,                                //第一隐含层第一个节点输入
           s2 = x*p2+p4,                                //第一隐含层第二个节点输入
           h1 = Sigmoid(s1),                            //第一隐含层第一个节点转换输出值
           h2 = Sigmoid(s2),                            //第一隐含层第二个节点转换输出值
           v1 = h1*p5+h2*p8+p11,                         //第二隐含层第一个节点输入
           v2 = h1*p6+h2*p9+p12,                         //第二隐含层第二个节点输入
           v3 = h1*p7+h2*p10+p13,                        //第二隐含层第三个节点输入
           f1 = Sigmoid(v1),                            //第二隐含层第一个节点转换输出值
           f2 = Sigmoid(v2),                            //第二隐含层第二个节点转换输出值
           f3 = Sigmoid(v3),                            //第二隐含层第二个节点转换输出值
           m1 = f1*p14+f2*p15+f3*p16+p17;             //输出层节点输入

Parameter p(17);
Variable x=[(x-xmin)/(xmax-xmin)], y=[0.1+0.8*(y-ymin)/(ymax-ymin)];
Function y = Sigmoid(m1);                           //输出层节点输出
Data;
x=1,7,13,19,25,31,37,43,49,55,61,67,73,79,85,91,97,103,109,115,121,127,133,139,145,151,157,163,169,175,1
81,187,193,199,205,211,217,223,229,235,241,247;
y=97.59,90.21,87.13,81.29,77.17,66.92,69.72,59.14,57.13,57.43,52.32,53.29,60.15,67.77,80.13,93.47,119.62,1
29.72,133.83,122.03,103.45,92.50,76.13,79.74,87.17,93.69,88.12,72.66,52.80,39.28,20.27,17.26,12.33,11.76,1
4.27,9.73,9.02,10.23,12.45,8.39,10.32,8.98;
```

结果：

```
Sum Squared Error (SSE): 0.00444540275176568
Root of Mean Square Error (RMSE): 0.0102879989629222
Correlation Coef. (R): 0.998986374756734
R-Square: 0.997973776949601
Adjusted R-Square: 0.997869868075222
Determination Coef. (DC): 0.997973698679974
Chi-Square: 0.00649567922202487
F-Statistic: 769.217841066365
```

Parameter	Best Estimate
p1	-15.9860223417208
p2	-3.64205509974063
p3	6.40605324358496
p4	3.7739409716722
p5	-5.01711708546069
p6	19.3157801906186
p7	2620.74926075723
p8	30.4013551850867
p9	-94.0919193883214
p10	-4912.09677998439
p11	-31.7273113493568
p12	79.2515559822619
p13	3775.19090692342
p14	2875.46483619027
p15	11.2438235794699
p16	-0.358038073640339
p17	-12.722509639272

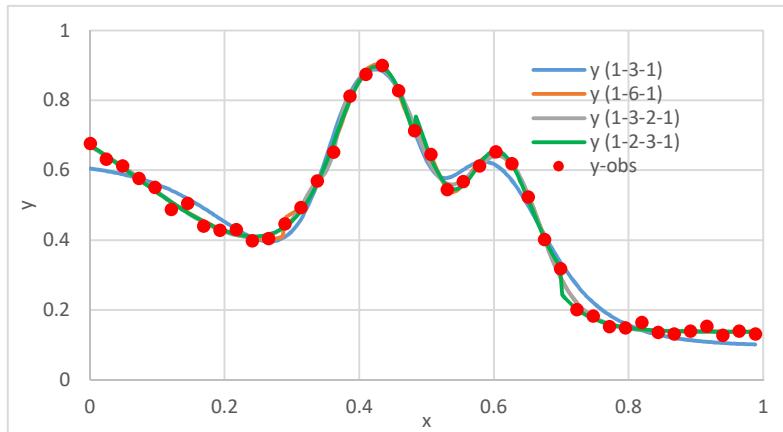


图 25-14 拟合问题 3 结果对比图

总体计算效果还是很好的，尤其是“1-2-3-1”和“1-6-1”两种神经网络结构结果最好，“1-3-1”结构参数最少，效果也最差。

前面展示的实例输出，不论计算还是实际值都是对应均匀化后的数值，实际应用中需要通过还原步骤进行数据复原。

25.5 小结

人工神经网络可以简单地归纳为一优化求解问题，对于小规模（连接权重数较少）的神经网络问题，可以通过 1stOpt 平台轻松实现，并且基于 1stOpt 强大的全局优化算法，可获得更好的结果，但对于大规模神经网络问题，在目前算力下，全局优化算法的高费时并不适用，尤其是当今的深度学习框架，网络隐含层成百上千，连接权重数高达几十上百亿，普通的算法很难满足要求。

神经网络有时被称为“黑箱”模型，因为只管输入输出，中间过程无法而知，通过本文实际案例可知，这是对神经网络的误解，神经网络有清晰的计算步骤和明确的计算公式，规模不大时甚至可以用一个表达式表示，只是当规模较大时公式规模过于庞大复杂而难以简单输出表示而已。

前面给出的代码只需更换数据以及对应的各数据变量的最大最小值（用于数据均一化），即可用于任意普通的拟合问题。