Guide of Matlab

内容概要： 数学建模算法

创建时间： 2022/4/7 13:41 **更新时间：** 2022/4/17 15:27

作者： **TwinkelStar**

神经网络工具箱
Neural Network Tools

1、操作系统相关环境

1）硬件环境：

- 电脑

2）软件环境：

- Matlab2018a(程序设计软件)

3）操作系统(2 选 1)：

- Windows7
- Windows10

2、BP 神经网络

神经网络在特征提取和建模上都有着相较于浅层模型显然的优势。深度学习善于从原始输入数据中挖掘越来越抽象的特征表示，而这些表示具有良好的泛化能力。不仅仅在数学建模中有着广泛的应用，在目标检测、计算机视觉、自然语言处理等领域成效卓然。

1）基本原理

生物神经系统是一个有高度组织和相互作用的数量巨大的细胞

组织群体。神经细胞也称为神经元，是神经系统的基本单元，他们按照不同的结合方式构建了复杂的神经网络。而我们人工神经网络就是模拟生物的神经网络，通过模拟出像生物神经网络那样，使得我们的智能体具有学习、记忆和认知等各种智能，图 1 为最基础全连接网络，只含有一个输入层，一个隐藏层，一个输出层的 BP 神经网络：

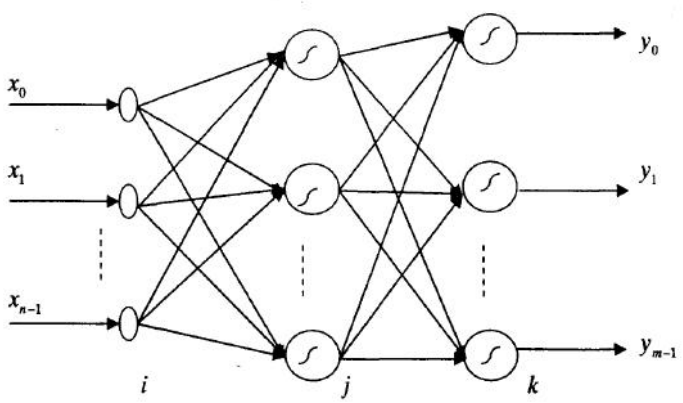


图 1.BP 神经网络的拓扑结构
fig1.BP Neural Network and Topology

但就针对数学建模而言，复杂的神经网络并不是都能用在建模中，我们可以搭建一些简单的网络求解问题。

BP 神经网络的学习过程是信号的正向传播（图像也是一种信号）与误差的反向传播两个过程组成。正向传播时，就像图 1 所示，由 x_0 作为输入，输入信号从输入层经过隐藏层，最后传向输出层，输出 y_0 在输出端产生输出信号。如果在输出层不能得到期望值，则转入误差信号反向传播，网络的权值由误差反馈调节，通过权值的不断迭代更新，最终修正网络的实际输出更接近期望输出，图 2 为算法流程图：

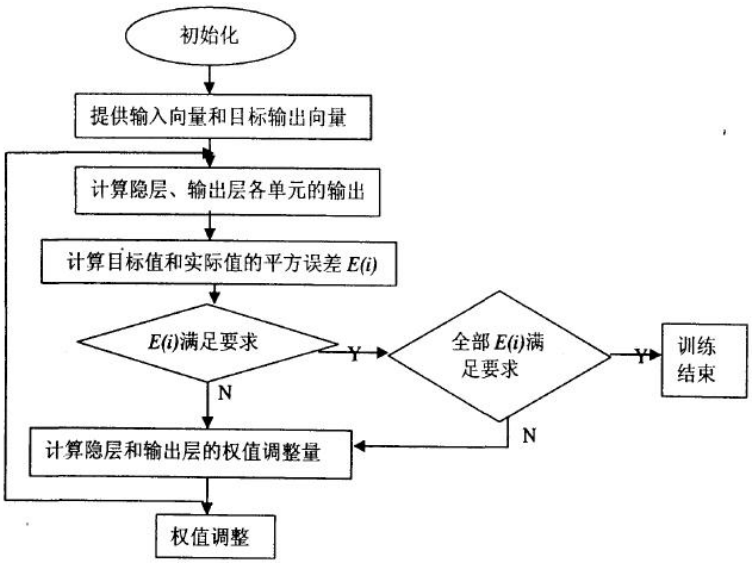


图 2.BP 神经网络流程图
fig2.BP Neural Network Charts

2）BP 算法步骤

第一步：权值初始化:w_{sq}=Random(")，sq 为 ij, jk;

第二步：依次输入 P 个学习样本。设当前输入为 P 个样本;

第三步：依次计算各层的输出:x_j，y_k，j=0，l，2，...，n_1，k=0，l，2，...，m-1。

第四步:求各层的反传误差:

$$\delta_{jk}^{(p)} = (d_k^{(p)} - y_k^{(p)}) y_k^{(p)} (1 - y_k^{(p)}), k = 0,1,\cdots,m-1$$

$$\delta_{ij}^{(p)} = \sum_{k=0}^{m-1} \delta_{jk}^{(p)} w_{jk}' x_j^{(p)} (1 - x_j^{(p)}), \quad j = 0,1,\cdots,n_1$$

图 3.BP 神经网络算法步骤
fig3.BP Neural Network Algorithm Steps

第五步：记录已学习过的样本数 P。如果 p<P，转到第二步继续计算；如果 p=P，转到第六步；

第六步：按权值修正公式修正各层的权值和阂值；

第七步：按新的权值再计算，若达到最大学习次数。则终止学习。否则转到第二步，继续新一轮的学习。

3、神经网络工具箱

因为从零开始搭建网络的过程及其复杂，数学建模时间有限，我们使用一些开源的工具可以更快的搭建网络模型，得到实验结果，以 Matlab 中的神经网络工具箱为例，在此之前，先介绍一下搭建神经网络的基本流程。

1）数据集的划分

将需要训练数据集划分为：训练集，测试集，真实预测集，以一组工业数据为例：

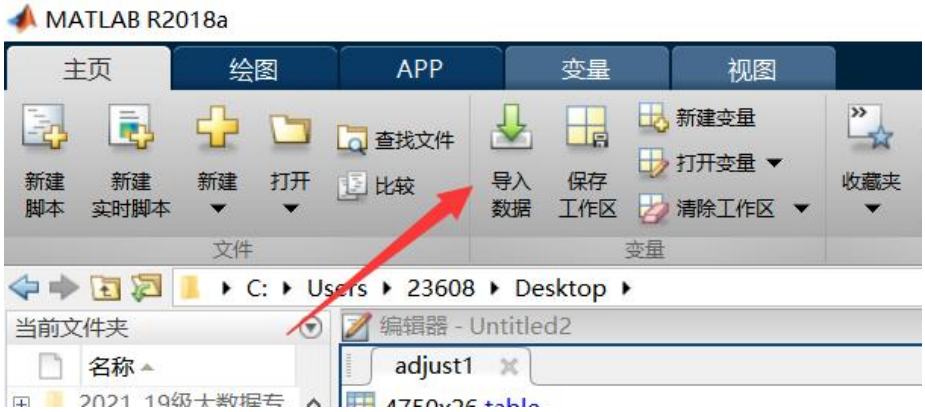


图 4.在 Matlab 中导入数据
fig4.Import Data of Matlab

该工业数据的第一列到第十八列为训练集，第十八列到第二十六列为真实预测集，通过拟合前十八列数据，以第十八列到第二十六列作为评价指标进行拟合预测：

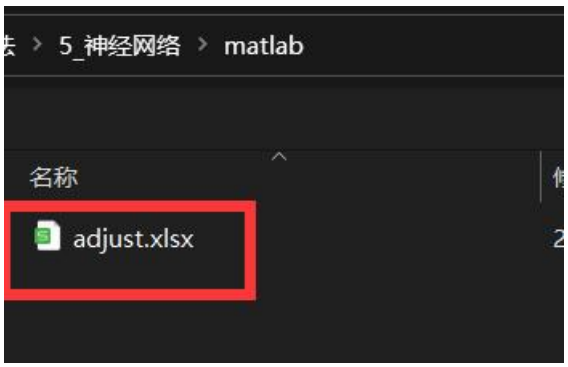


图 5.选择数据表
fig5.Selection Data Table

导入数据之后，勾选导入所选内容，将数据表作为变量存储在 Matlab 中如图 6 所示：

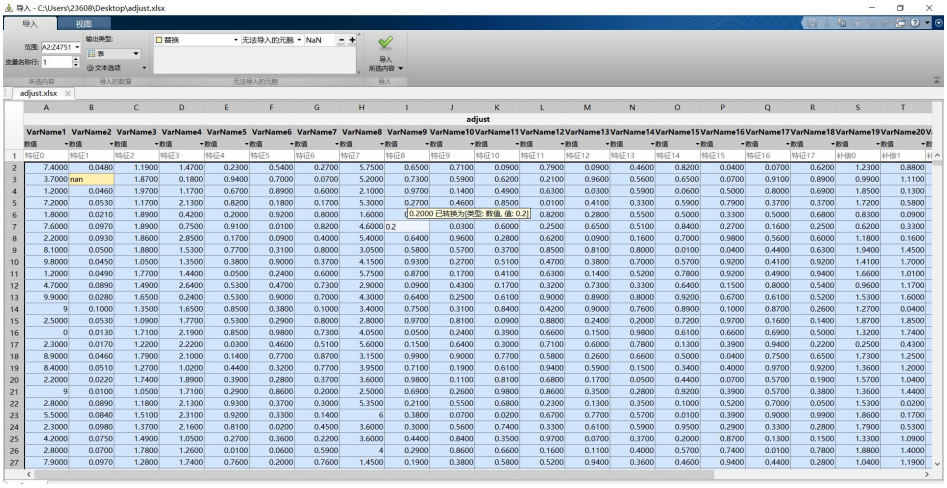
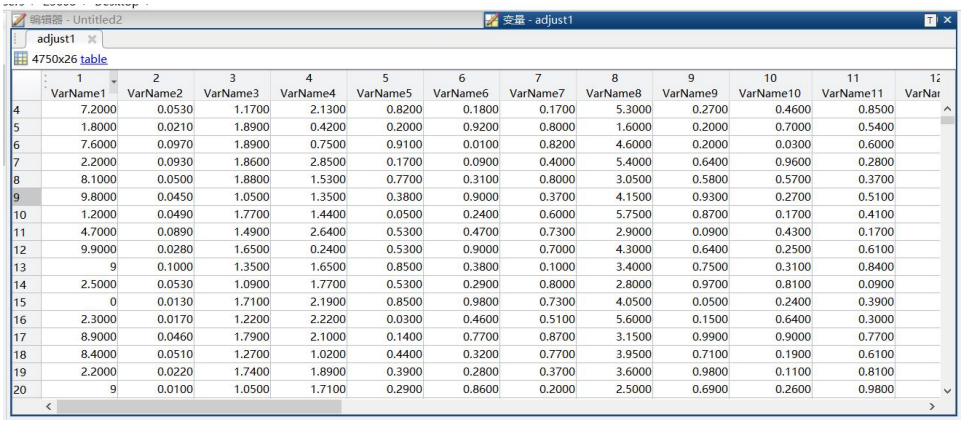


图 6.选择数据表导入
fig6.Selection Data Table Import

在 matlab 的变量区打开 adjust 变量，数据如图 7 所示：



	1 VarName1	2 VarName2	3 VarName3	4 VarName4	5 VarName5	6 VarName6	7 VarName7	8 VarName8	9 VarName9	10 VarName10	11 VarName11	12 VarNar
4	7.2000	0.0530	1.1700	2.1300	0.8200	0.1800	0.1700	5.3000	0.2700	0.4600	0.8500	
5	1.8000	0.0210	1.8900	0.4200	0.2000	0.9200	0.8000	1.6000	0.2000	0.7000	0.5400	
6	7.6000	0.0970	1.8900	0.7500	0.9100	0.0100	0.8200	4.6000	0.2000	0.0300	0.6000	
7	2.2000	0.0930	1.8600	2.8500	0.1700	0.0900	0.4000	5.4000	0.6400	0.9600	0.2800	
8	8.1000	0.0500	1.8800	1.5300	0.7700	0.3100	0.8000	3.0500	0.5800	0.5700	0.3700	
9	9.8000	0.0450	1.0500	1.3500	0.3800	0.9000	0.3700	4.1500	0.9300	0.2700	0.5100	
10	1.2000	0.0490	1.7700	1.4400	0.0500	0.2400	0.6000	5.7500	0.8700	0.1700	0.4100	
11	4.7000	0.0890	1.4900	2.6400	0.5300	0.4700	0.7300	2.9000	0.0900	0.4300	0.1700	
12	9.9000	0.0280	1.6500	0.2400	0.5300	0.9000	0.7000	4.3000	0.6400	0.2500	0.6100	
13	9	0.1000	1.3500	1.6500	0.8500	0.3800	0.1000	3.4000	0.7500	0.3100	0.8400	
14	2.5000	0.0530	1.0900	1.7700	0.5300	0.2900	0.8000	2.8000	0.9700	0.8100	0.0900	
15	0	0.0130	1.7100	2.1900	0.8500	0.9800	0.7300	4.0500	0.0500	0.2400	0.3900	
16	2.3000	0.0170	1.2200	2.2200	0.0300	0.4600	0.5100	5.6000	0.1500	0.6400	0.3000	
17	8.9000	0.0460	1.7900	2.1000	0.1400	0.7700	0.8700	3.1500	0.9900	0.9000	0.7700	
18	8.4000	0.0510	1.2700	1.0200	0.4400	0.3200	0.7700	3.9500	0.7100	0.1900	0.6100	
19	2.2000	0.0220	1.7400	1.8900	0.3900	0.2800	0.3700	3.6000	0.9800	0.1100	0.8100	
20	9	0.0100	1.0500	1.7100	0.2900	0.8600	0.2000	2.5000	0.6900	0.2600	0.9800	

图 7.成功导入数据表
fig7.Successful Data Table Import

编辑脚本函数，将数据集划分为训练集和真实数据集，方便使用神经网络工具箱对数据集进行拟合，如图 8 所示：

```
import = table2array(adjust1);%将表格数据转化为数组
trian = import(:,1:18);%划分训练集
test = import(:,18:26);%划分真实数据集
```

图 8.划分数据集
fig8.Divide Data Table

2）搭建网络

选择 APP 选项，APP 是 Matlab 工具箱的集合，里面包含了大量的数据分析工具，如机器学习、数据分析与统计、信号处理、仿真控制、计算机视觉处理等工具包，非常适合新手快速上手掌握，打开“Neural Net Fiting”工具，如图 9 所示：

图 9.打开 Matlab 工具包
fig9.Open Matlab Tools

选择 Next 选项，进入下一步，如图 10 所示：

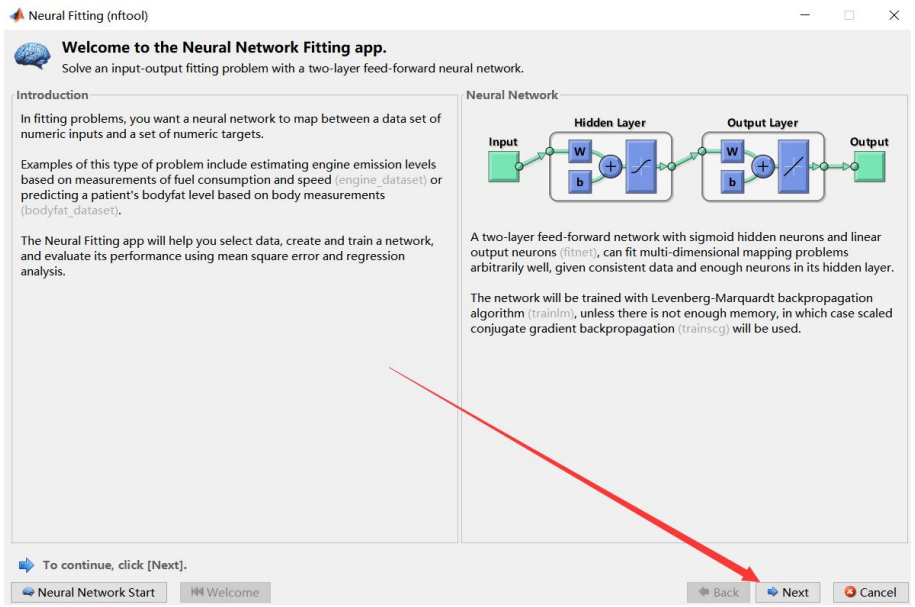


图 10.选择 Next 选项
fig10.Selection Button of Next

Inputs：表示的是要喂入网络的数据，也就是我们划分的训练集 trian，选择变量区的 trian 变量传入网络。

Targets：表示的是要预测的数据，也就是我们划分的真实数据集，作为网络的输出。

选择好数据之后点击 Next，如图 11 所示：

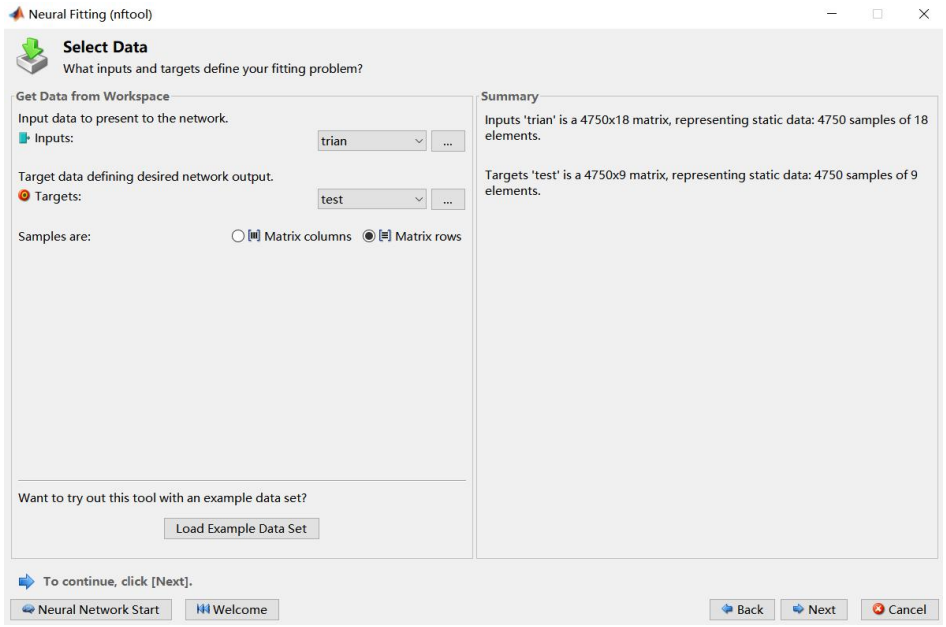


图 11.选择数据

fig11.Selection Data

在使用训练集对参数进行训练的时候，人们通常会将一整个训练集分为三个部分，一般分为：训练集（train_set），评估集（valid_set），测试集（test_set）这三个部分。这是为了保证训练效果而特意设置的。其中测试集很好理解，其实就是完全不参与训练的数据，仅仅用来观测测试效果的数据。而训练集和评估集则牵涉到下面的知识。

因为在实际的训练中，训练的结果对于训练集的拟合程度通常还是挺好的（初始条件敏感），但是对于训练集之外的数据的拟合程度通常就不那么令人满意了。因此我们通常并不会把所有的数据集都拿来训练，而是分出一部分来（这一部分不参加训练）对训练集生成的

参数进行测试，相对客观的判断这些参数对训练集之外的数据的符合程度。这种思想就称为交叉验证（Cross Validation）。

在 Matlab 中已经为我们划分好了训练集、验证集、测试集，无需再做修改，选择 Next 进行下一步，如图 12 所示：

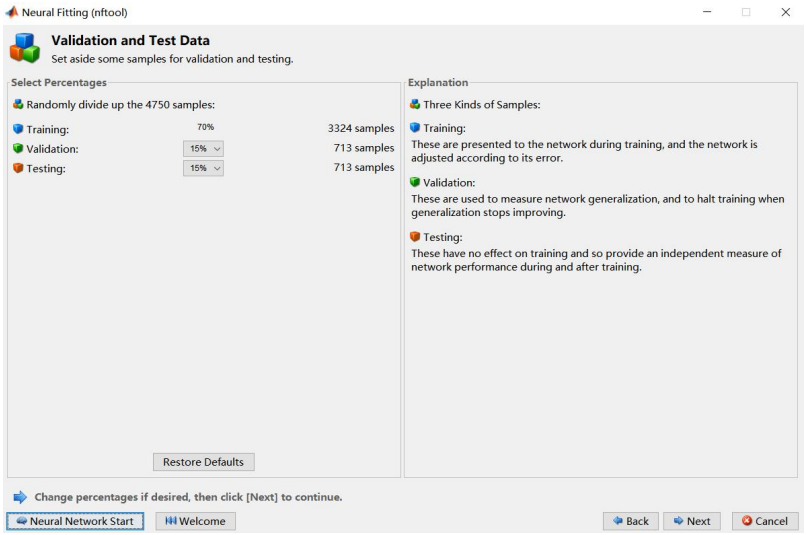


图 12.交叉验证

fig12.Cross Validation

神经网络的搭建，可以设置隐藏层的层数，从图 13 中可以看出，我们的 Input 是我们的神经网络的输入，Output 输入的参数的维度为 18，输出的参数为 9，至此，我们搭建好了全连接网络。

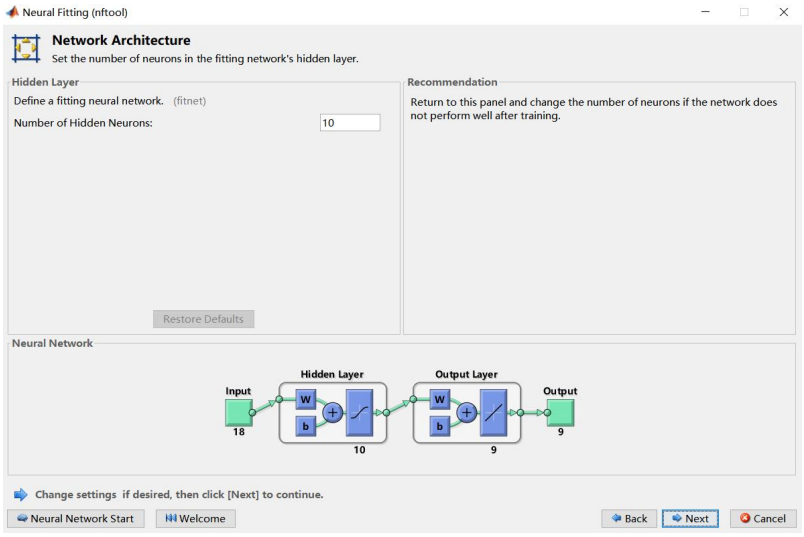


图 13.BP 网络模型

fig13.BP Network Model

选择其中一种算法进行训练，不同的算法训练有不同的效果，LevenberaMarauardt 算法通常需要更多的内存，但需要更少的时间。当泛化停止改进时，训练将自动停止，这可以从验证样本的均方误差的增加中看出。默认即可，选择 Train 进行训练，如图 14 所示：

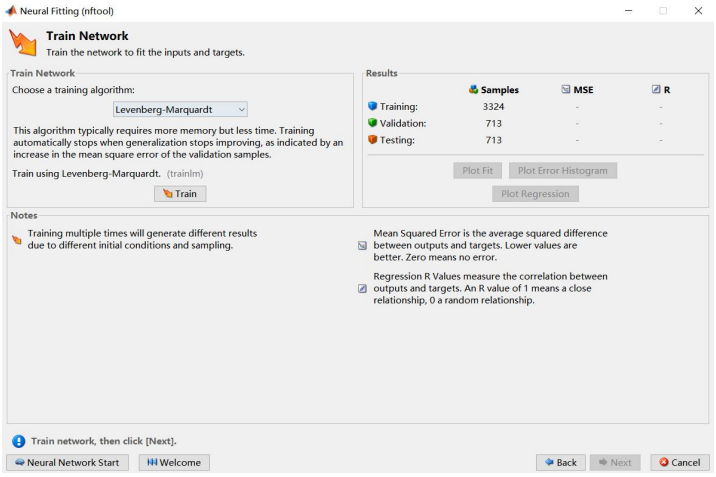


图 14.BP 网络模型训练
fig14.BP Network Model Trian

3）训练模型

训练好模型之后会弹出训练结果可视化窗口，包含训练结果、误差分析，回归分析等，如图 15 所示：

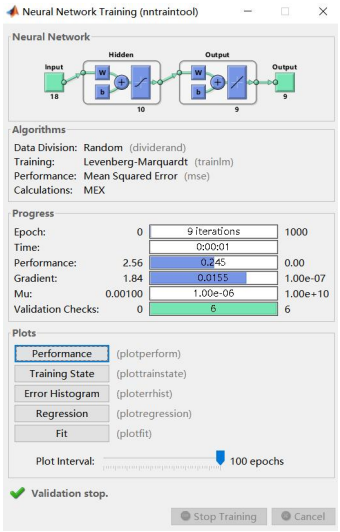


图 15.BP 网络模型训练可视化页面
fig15.BP Network Model Trian Charts

4）模型评估

模型的性能评估，反应的是模型在哪个阶段 loss 值收敛到最小；交叉验证可以作为评价模型的鲁棒性是否良好的依据；误差分析则是分析模型的预测水平如何。在数学建模中，我们也是需要对实验结果进行反复的测度，用不同的验证方法去证明我们的实验结果，通过多组不同的对比实验，完成对模型的评估和分析，如图 16 所示：

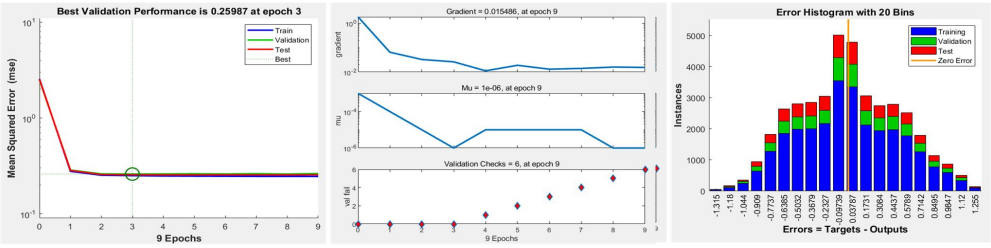


图 16.性能评估(左).交叉验证(中). 误差分析(右).
fig16.BP Network Model Trian Charts

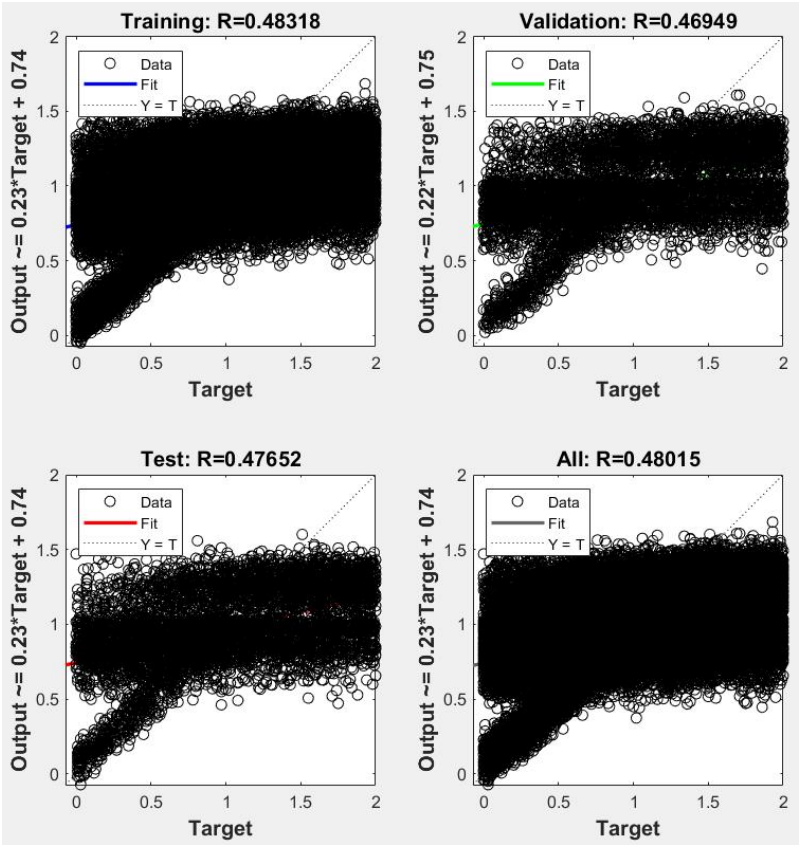


图 17.非线性回归
fig16.Nonlinear Regression