

基于神经网络的简支梁分界面检测模型

小组成员：吴秉宪 潘浩然 李鸣鹤

一、研究目的

在工程实践中，结构安全是至关重要的考虑因素。然而，建筑材料中可能存在各种杂质，这些杂质可能导致结构不同部分的强度存在差异，这种差异可能在某些特定部位更为显著，从而增加了结构在这些区域发生损坏的风险，造成安全隐患。为了全面评估和确保结构的整体安全性，有必要进行分界面检测，以确定结构中存在的不同材料的位置和性质。分界面检测能够提供对结构内部材料异质性的详细了解，不仅有助于识别可能存在的结构薄弱点，还为工程师提供了在设计和维护过程中优化结构性能的机会。

然而目前，工程结构中各位置的物理性质主要依靠传统的手动测量和视觉检查方法，这些方法不仅耗时耗力，而且效率低下，精度也难以满足日益提高的工程质量要求。立足于解决有实际意义的力学问题，力求原创地应用课堂讲授的知识解决微小问题，用科学促进工程，我们最终决定尝试使用神经网络模型来探索这类工程问题的解法。

为了简化问题，化繁为简，我们将问题限制到一个简支梁上，考虑两种材料的分界面，选用梁的挠度作为特征量，来研究分界面位置与挠度分布的关系。在这里我们并不需要考虑不同类型荷载下的挠度分布，因为在检测过程中，外荷载是可以人为定义的，因此在本项目中默认使用均布荷载下的挠度数据作为模型输入。于是这个问题本质上成为了一类回归问题，这是神经网络模型的看家本领之一，本项目旨在开发一个基于神经网络的简支梁分界面检测模型，以解决在现有的检测方法中存在的局限性。

二、理论支撑

在我们的项目中，我们的目标是构建一个神经网络模型，输入简支梁的挠度分布，令其预测两种材料分界面的位置。这个任务的理论基础是经典的材料力学弯曲理论，即梁在外力作用下的挠度与外力分布及其物理特性（如杨氏模量、截面惯性矩）之间的关系。

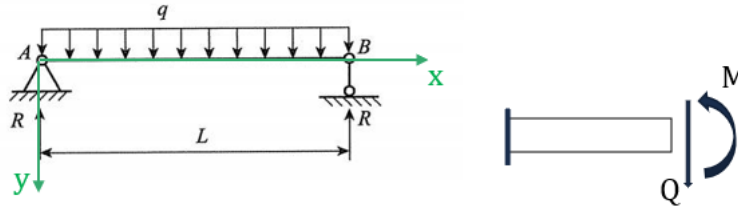


图 1 理论推导的正方向规定

由弯曲理论，简支梁的挠度 $v(x)$ 、转角 $\varphi(x)$ 、剪力 $Q(x)$ 和弯矩 $M(x)$ 之间存在密不可分的联系。首先规定正方向如图 1 所示，根据微元体的受力平衡，我们可以推得：

- $\frac{dv}{dx} = \varphi(x)$
- $\frac{d\varphi}{dx} = -\frac{1}{EI}M(x)$
- $\frac{dM}{dx} = Q(x)$

$$4. \frac{dQ}{dx} = -q(x)$$

将这些关系整合起来，我们得到一个四阶微分方程：

$$EI \frac{d^4 v}{dx^4} = q(x)$$

它直接描述了挠度和外荷载的关系，若给定边界条件，简支梁的挠度就可以直接由弯曲刚度 EI 和分布力 $q(x)$ 决定。

进一步地，我们经过理论推导，得出在均布载荷 q 下，具有两种材料的简支梁挠度计算公式如下，从中可以看出，截面位置 a 与挠度分布 $v(x)$ 有着复杂的非线性关系，这也证明了，这个问题是可以用神经网络模型求解的。

$$v(x) = \begin{cases} \frac{q}{EI_1} \left(\frac{lx^3}{12} - \frac{x^4}{24} + \left(-\frac{la^2}{4} + \frac{a^3}{3} - \frac{a^4}{8l} \right) x \right) + \frac{q}{EI_2} \left(\frac{la^2}{4} - \frac{a^3}{3} + \frac{a^4}{8l} - \frac{l^3}{24} \right) x, & 0 < x < a \\ \frac{q}{EI_1} \left(\frac{a^4}{8} - \frac{la^3}{6} + \left(\frac{a^3}{6} - \frac{a^4}{8l} \right) x \right) + \frac{q}{EI_2} \left(\frac{lx^3}{12} - \frac{x^4}{24} + \frac{la^3}{6} - \frac{a^4}{8} + \left(-\frac{l^3}{24} - \frac{a^3}{6} + \frac{a^4}{8l} \right) x \right), & a < x < l \end{cases}$$

与此同时，上述解析解已经考虑了简支梁的边界条件和实际载荷的情况，为我们的神经网络模型打下了生成训练数据和测试数据的基础，**根据梁挠度的解析解，我们预先编写程序计算在不同分界面位置下，梁结构各个位置的挠度值以作为模型的输入，并保留分界面位置作为输出，得到了数据集。**

最终，我们的任务就是将这些理论方程和解析解转化为神经网络模型的训练和预测过程，让模型学习如何根据输入的挠度分布 $v(x)$ 来预测截面位置 a 。

三、模型结构与代码介绍

上面已经提到，本项目的核心问题实质上是一个回归问题，因此前馈神经网络与该问题比较适配，我们最终选取了多层感知机（Multilayer Perceptron, MLP）作为我们搭建模型的基本方向，在此基础上做出微调。在本节中将展示最终代码的模型结构，并对代码做一个简要的介绍，训练代码过程中做出的具体调整将在下一节中叙述。

首先介绍数据集的生成方式。我们首先假定简支梁的长度固定为 10 米，外荷载固定为 10kN/m。数据集由 100000 条数据组成，每一组数据由 1004 个值组成，前三个值为待预测值，分别为分界面位置、材料一的弯曲刚度和材料二的弯曲刚度，后面的 1001 个值为将 10 米长的简支梁划分为 1000 等分点处的挠度。其中，分界面位置在 0~1000(cm)中随机生成，两种材料的弯曲刚度分别在 $1e11 \pm 2e10$ 和 $2e11 \pm 2e10$ 的范围内随机生成。而离散的挠度分布将通过第二节中的公式以及分界面位置和两种材料的弯曲刚度来生成。

下面是神经网络模型结构的介绍。模型选取数据中的 1001 个值中的 250 等分点处的值作为一个特征量（即每四个值中选一个特征），因此输入数据是 250 维的，这样做是为了相对降低输入的维数，我们发现这样选取特征在拟合过程中会更加容易收敛。模型采用了四层网络结构，如下图 2、图 3 所示（图中的神经元个数经过了放缩，仅作参考），其中四层网络的神经元个数递减，分别为 150、100、50 个，并最终输出三维结果。此处输出三维结果，是因为在我们实践过程中，发现除了分界面位置可以预测外，通过大量数据的训练，模型还具有预测两种材料的弯曲刚度的能力，于是我们对数据集进行了重新生成，并最终训练出一个能够同时预测分界面位置和材料弯曲刚度的模型。

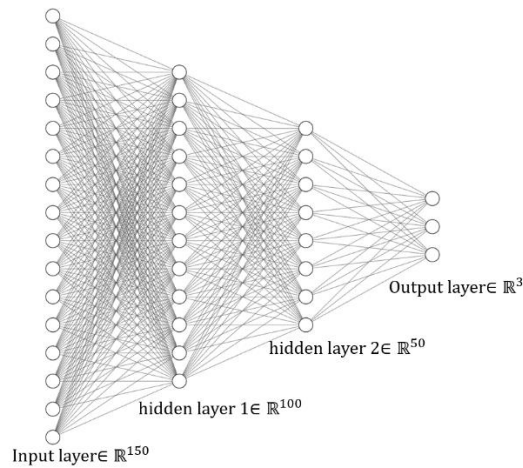


图 2 模型结构示意图

Layer (type)	Output Shape	Param #
Linear-1	$[-1, 1, 150]$	37,650
Linear-2	$[-1, 1, 100]$	15,100
Linear-3	$[-1, 1, 50]$	5,050
Linear-4	$[-1, 1, 3]$	153
Total params: 57,953		
Trainable params: 57,953		
Non-trainable params: 0		
Input size (MB): 0.00		
Forward/backward pass size (MB): 0.00		
Params size (MB): 0.22		
Estimated Total Size (MB): 0.22		

图 3 模型参数

在训练时，我们选取了处理回归问题常用的均方误差（MSE）作为损失函数。同时，在比较了 Adam 和随机梯度下降（SGD）的效果后，我们最终选择了 SGD 作为优化器，这也是在汇报之后我们根据实际情况做出的一个调整。由于最初选用的 10 个 epochs 不足以达到比较好的效果，我们将 epochs 的个数扩充到了 50 个。此外，学习率的设置是我们最头疼的一点，由于数据的差值很小，如果学习率过高，就容易导致梯度爆炸或是模型发散无法拟合，学习率过低则收敛速度比较缓慢，经过多次测试后我们最终选择了 0.0002 作为学习率。

下面是我们代码过程的简要介绍，在代码中也有相应的注释便于理解。

首先需要进行必要的环境配置，并进行数据的读取与预处理。我们导入了 pytorch 和 pandas 等关键的 Python 库用于搭建神经网络和处理数据。在读取数据时，我们先对读入的数据做了检查，利用 pandas 提取数据的特点（如最小值、最大值等），确保我们生成的数据在我们的预期之内，例如，在生成数据时，我们规定两种材料的弯曲刚度应当在 $\pm 2e10$ 的范围内波动，通过检查最小值和最大值就可以确定此处不出纰漏。接下来是数据集的划分，我们以 7:3 的比例对数据集进行了随机划分，将数据集分成训练集和测试集，并以 32 组数据为一个 batch 进行分组。

随后，我们通过 pytorch 定义了网络结构和模型超参数，定义了如上所述的四层网络结构和反向传播算法，选取了 MSE 作为损失函数、SGD 作为优化器，学习率设置为 0.0002。接

下来，我们利用训练集进行 50 个 epochs 的模型训练，记录训练过程中损失函数的变化情况并可视化，发现损失函数随着 epochs 的增加能够很快下降，收敛良好，如图 4 所示。

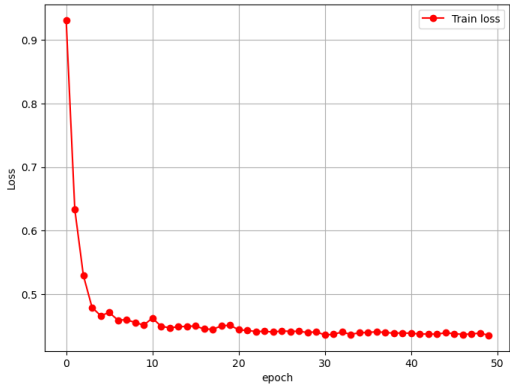


图 4 损失函数的变化趋势

最后，我们进行了结果的评估。结果评估分为两部分，利用测试集进行测试和利用外来数据来进行测试。其中，我们将测试集上的测试结果也做了可视化，分别作出了分界面位置和两种材料弯曲刚度的残差图，如图 5、图 6、图 7 所示，从图中可见，预测的效果良好。

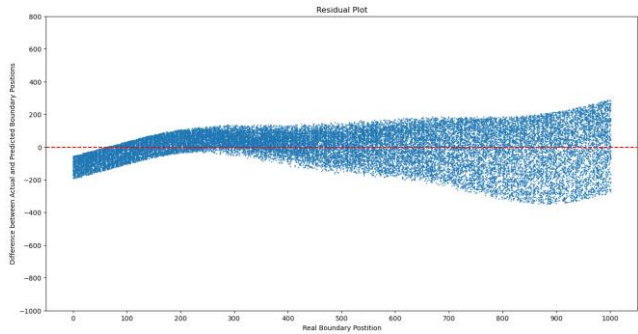


图 5 分界面位置的残差图

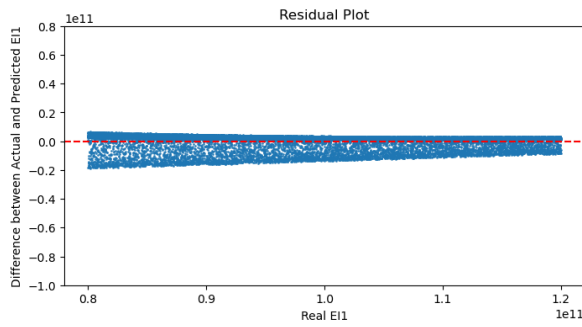


图 6 材料一弯曲刚度的残差图

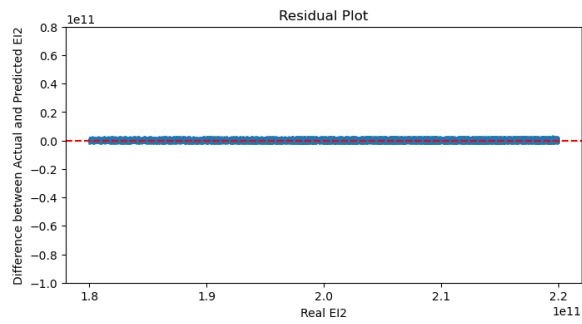


图 7 材料二弯曲刚度的残差图

四、参数调整

在优化神经网络的过程中，我们遇到了一些阻碍，并根据反馈不断调整模型的参数，最终取得了一个比较好的效果。

在最开始时，我们的目标是在固定两种材料的弯曲刚度的前提下，训练一个仅检测分界面位置的神经网络。同时，最初我们认为特征数量过多时，如果数据量不够大，就会出现欠拟合的情况出现，因此我们起初选择只取 3 个特征值作为模型输入，分别取了四分之一点处的挠度、四分之三点处的挠度和最大挠度值作为特征。值得一提的是，根据作业和经验，我们选取 0.1 作为初始的学习率，发现完全无法拟合，输出的数据与实际数据相去甚远，经过长时间的试验，我们发现了这个问题，并将学习率降低了几个量级。令 epochs=30，学习率固定为 0.00001，得到了如下的网络模型、损失函数变化以及分界面位置残差图如下图 8~10。

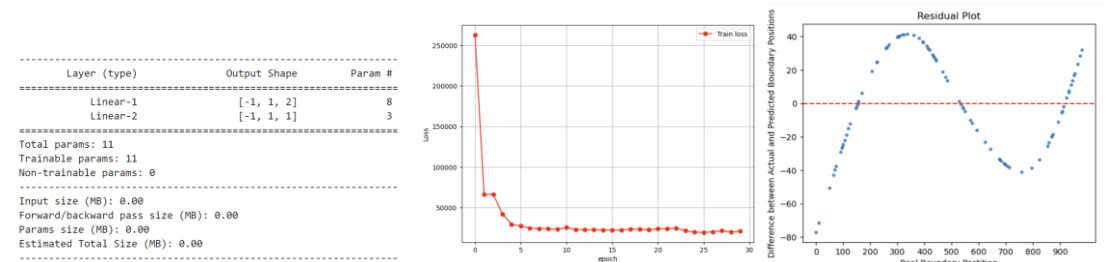


图 8 最初的两层网络结构 图 9 损失函数变化图 图 10 分界面位置残差图

随后我们添加了两层隐藏层，结果出现了参数为 nan 的问题，经检验是因为数据的差值很小，学习率没有调整可能导致梯度爆炸导致的，于是重新将学习率下调为 0.000001，得到了图 11~13 的结果。

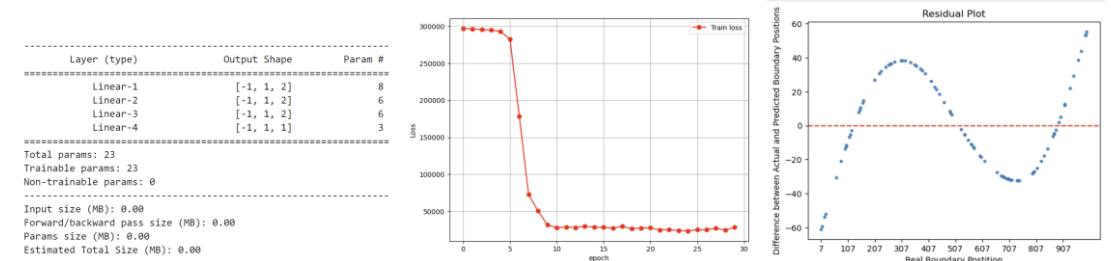


图 11 包含隐藏层的四层网络结构 图 12 损失函数变化图 图 13 分界面位置残差图

为了比较稀疏特征与复杂特征的效果，我们在每四个挠度值中选取一个挠度为特征，将输入维度增加到 250 维，同时将学习率再缩小为原来的 1/10，得到图 14~16 的结果。

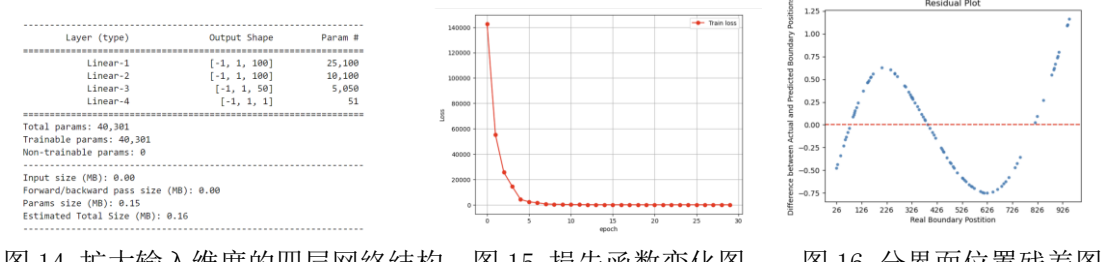


图 14 扩大输入维度的四层网络结构 图 15 损失函数变化图 图 16 分界面位置残差图

在此过程中，我们创造性地提出，尝试将两种材料的弯曲刚度也作为待预测值参与模型训练（这也实现了我们的预期拓展目标），于是修改了数据集的生成代码后生成了新的数据集，将模型完善为能够同时预测分界面位置、EI1 和 EI2 的网络。此处我们最先需要修改的自然 MLP 的输出维度，将输出维度扩大到三维，得到了图 17 所示的网络结构。随后我们在多次训练测试中都遇到了 nan 数值溢出问题，调整学习率也无济于事，然后意识到大概率是因为输出数据的值过大（ $10e11$ ）导致梯度爆炸，于是归一化到 1 附近，将问题成功解决。程序随后得以顺利运行，但是由于训练时数据量很小，结果自然是很不吻合，如图 18~19 所示。

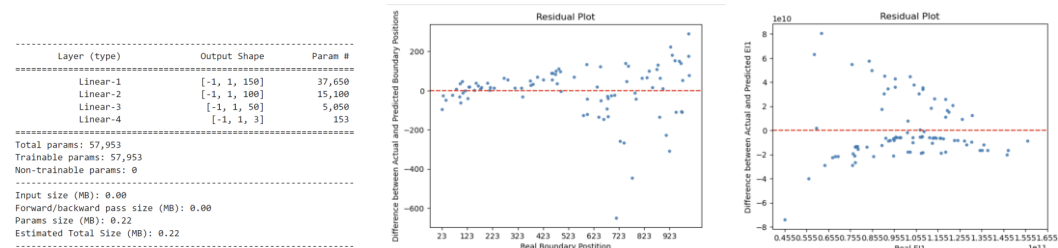


图 17 修改输出维度的网络结构 图 18 分界面位置残差图 图 19 EI1 残差图

在上面我们已经发现，将输入维度扩大后，效果并不会变差，与原来的效果近乎持平，于是我们决定大幅扩大数据量至 10000 条，然后仍然保持 250 维的输入量进行训练。其中，我们将学习率回调至 0.0001，得到了如图 20~22 的结果。

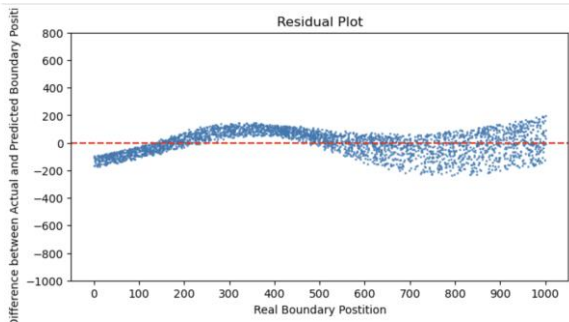


图 20 分界面位置残差图

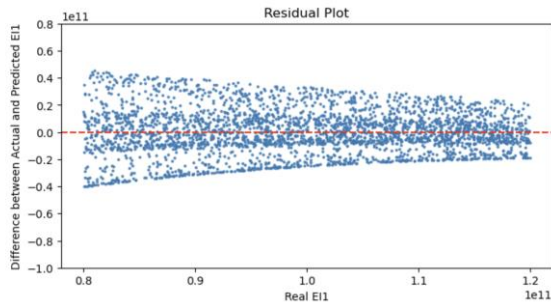


图 21 EI1 残差图

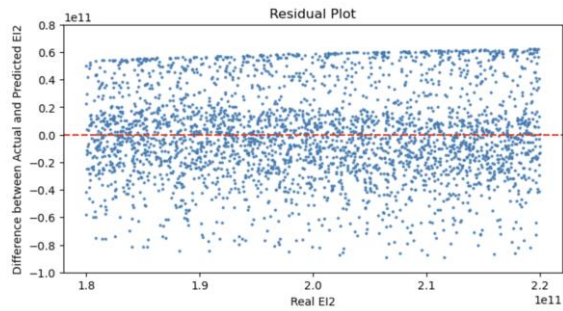


图 22 EI2 残差图

之前的数据量仍然较小，都是在 CPU 上运行的，随后我们在原代码中设置了 GPU 加速，并将数据加到 10w 条，并将学习率增加到 0.0002，epochs 增加到 50，发现拟合效果明显好了许多，同时运行时间缩减了不少，结果如图 23~26 所示。这也是我们最终所采用的模型。

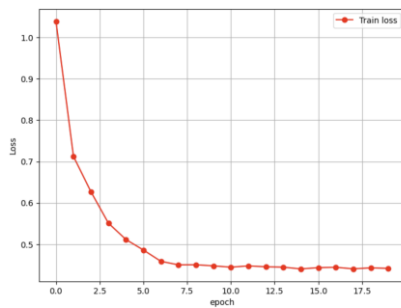


图 23 损失函数变化

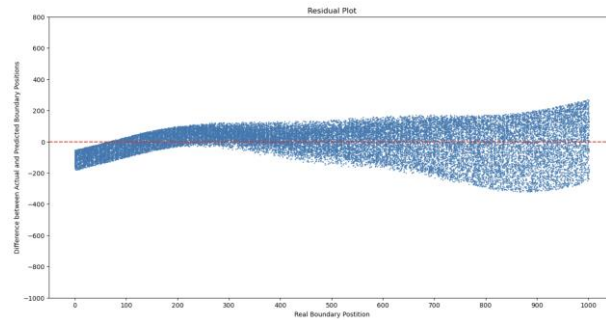


图 24 分界面位置残差图

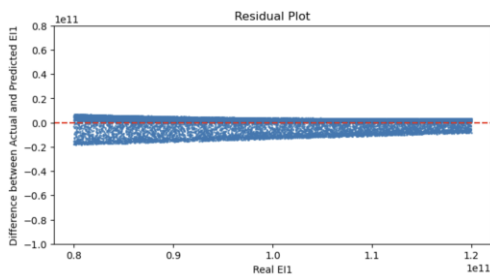


图 25 EI1 残差图

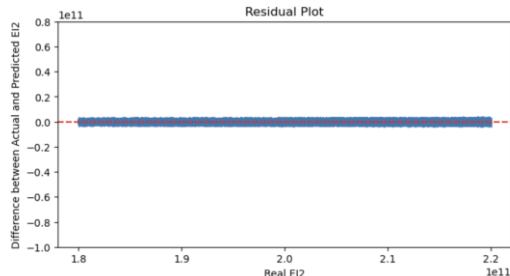


图 26 EI2 残差图

下面，我们还做出了一些别的尝试，但是发现效果都没有这个模型好。

我们尝试将所有挠度值都作为输入特征，并且加多了一层神经网络并增加了每层的神经元个数（如图 27），将学习率改为 0.00001，结果和原来（四层 MLP）相比，收敛的速度下降了（图 28），误差的范围增大了（图 29~30），而且耗费更多时间。

```

# 定义模型结构
def __init__(self, n_inputs):
    super(MLP, self).__init__()
    # 输入层
    self.hidden1 = Linear(n_inputs, 500)
    xavier_uniform_(self.hidden1.weight)
    # 第二层
    self.hidden2 = Linear(500, 300)
    xavier_uniform_(self.hidden2.weight)
    # 第三层
    self.hidden3 = Linear(300, 100)
    xavier_uniform_(self.hidden3.weight)
    self.hidden4 = Linear(100, 20)
    xavier_uniform_(self.hidden4.weight)
    # 输出层
    self.output = Linear(20, 3)

```

图 27 新加一层隐藏层

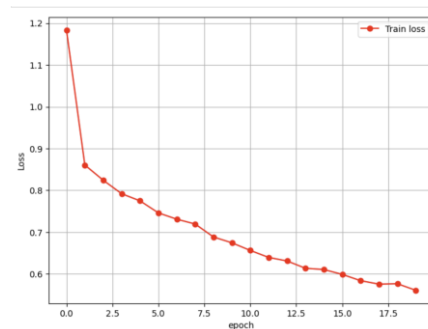


图 28 损失函数变化图

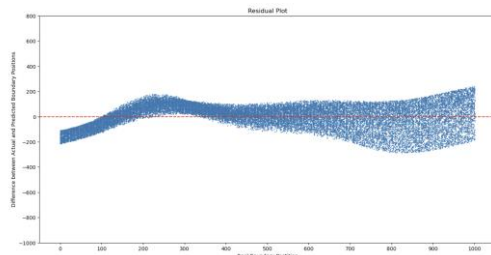


图 29 分界面位置残差图

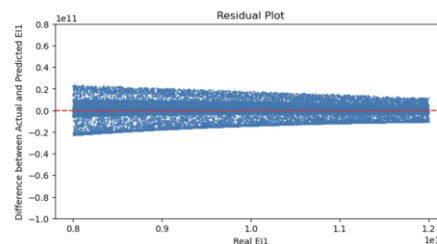


图 30 EI1 残差图

接下来输入数据降为 250 维，同时每一层的神经元个数也进行了一定压缩（图 31），但仍然保持了 5 层的网络，使用 GPU 运行，此时收敛速度加快了（图 32），结果有一定改善但并不比 4 层网络时好（图 33~35）。

```

cuda
-----
Layer (Type)          Output Shape         Param #
-----
Linear-1              [-1, 1, 300]         75,300
Linear-2              [-1, 1, 200]         60,200
Linear-3              [-1, 1, 100]         20,100
Linear-4              [-1, 1, 50]          5,050
Linear-5              [-1, 1, 3]           153
-----
Total params: 160,803
Trainable params: 160,803
Non-trainable params: 0
Input size (MB): 0.00
Forward/backward pass size (MB): 0.00
Params size (MB): 0.61
Estimated Total Size (MB): 0.62

```

图 31 修改后的网络结构

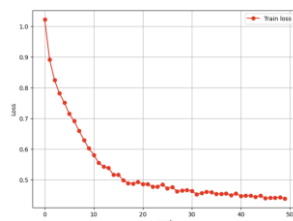


图 32 损失函数变化图

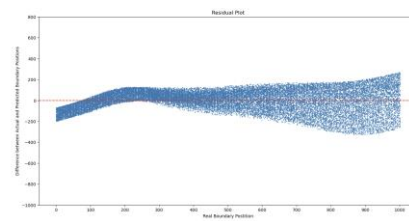


图 33 分界面位置残差图

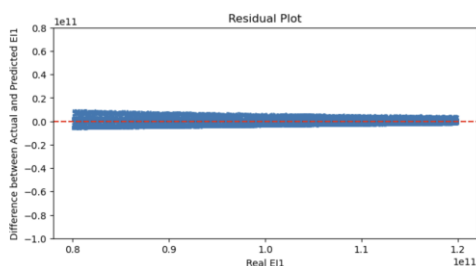


图 34 EI1 残差图

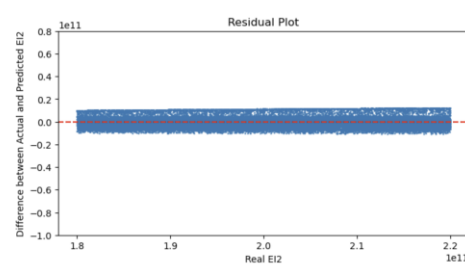


图 35 EI2 残差图

因此，总体来看 4 层 MLP 已经足够，我们最终选用了 4 层 MLP 的最佳参数设置作为最终模型。

五、改进空间与拓展方向

在训练模型的过程中，我们针对遇到的许多问题都做出了不错的改进（具体修改见第四

节)，我们的模型最终能够达到不错的收敛效果，分界面位置以及两种材料的弯曲刚度的预测都没有太大的偏差。但是在我们的调参过程中，始终无法根除的一个问题就是学习率的调整问题，我们也无法保证目前选取的学习率是最优的，因此在学习率的优化方面我们仍然具有一定的改进空间。

本项目的可拓展应用方向非常广泛。在汇报中我们曾经指出，本项目在特征参数、分界面个数和结构类型上都能够得到拓展。其中，我们已经实现了特征参数的拓展，将两种材料的弯曲刚度也作为预测目标，从而拟合出了能够适应各类材料的模型。在分界面个数方面，当前的模型以两种材料的分界面为研究对象，未来可以考虑拓展到由多种材料组成的梁结构的检测，拓展其适用性。此外，最重要的一个应用前景是，本项目为神经网络在各类结构的故障检测中的应用打下了一个基础，未来可以探索由简支梁向各类简单结构乃至复杂结构的拓展，训练出能够检测各类结构分界面的模型，有重要的工程意义。

附：小组成员分工

前期的准备工作中，理论公式的推导核验、数据集的生成和模型的设计搭建与训练调试均由吴秉宪、李鸣鹤、潘浩然三人共同完成，并线进行，集思广益。在后期的调试工作中，潘浩然和吴秉宪主要负责模型参数的调整和记录，李鸣鹤和吴秉宪负责了报告的撰写。此外，吴秉宪负责了 PPT 制作，潘浩然负责了汇报展示。