**Tensorflow Playground 试用报告**

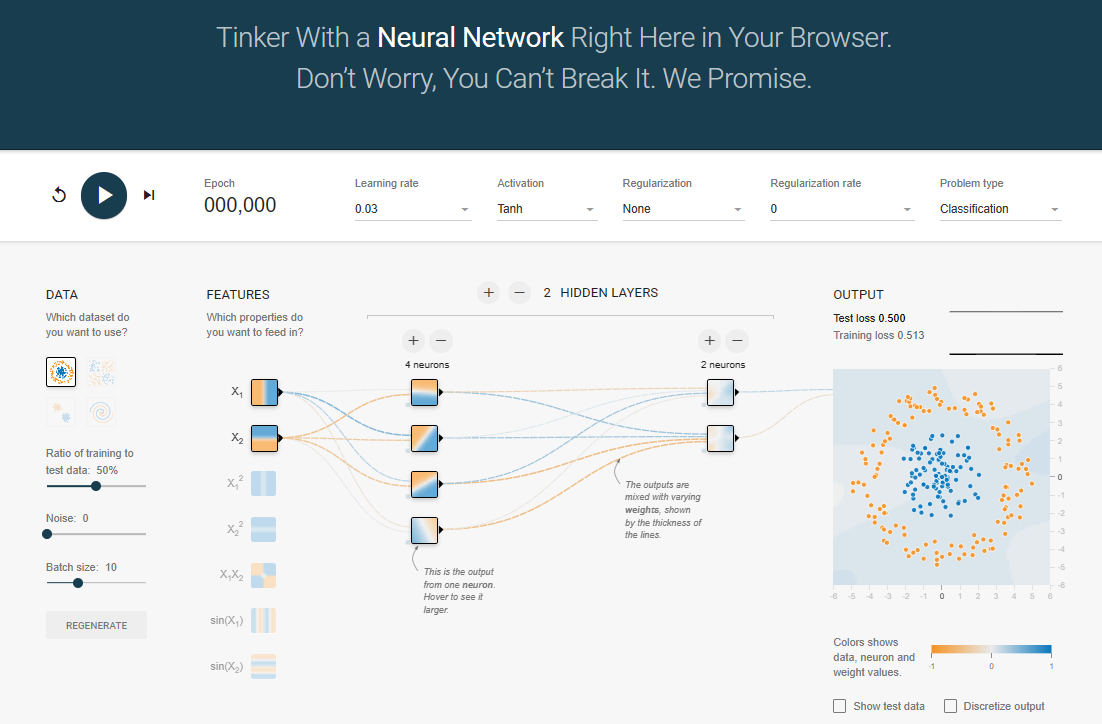
姓名：李智 学号：123106222804

1. **项目介绍**

TensorFlow Playground是一个在线的交互式可视化工具，用于学习和理解深度学习模型的基本概念和原理。它是由谷歌TensorFlow团队开发的，旨在通过简化和可视化深度学习模型的构建和训练过程，帮助初学者更好地理解神经网络的工作原理。

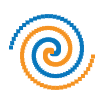
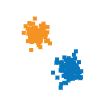
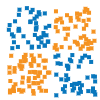
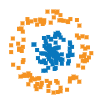
1. **使用说明**

PlayGround页面如图所示，模型主要分为DATA（数据），FEATURES（特征），HIDDEN LAYERS（隐含层），OUTPUT（输出层）以及页面上方的相关参数设置。

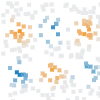


1. DATA（数据）：

对于分类问题，DATA栏里提供了4种不同形态的数据，分别是圆形、异或、高斯和螺旋。平面内的数据分为蓝色和黄色两类，其中第三类数据属于线性可分的类型，其它几类属于线性不可分。如图所示：



对于回归问题，DATA栏提供了2种形态的数据，前者线性可分，后者线性不可分，如图所示：

数据提供了三种配置，可以调节噪声、训练数据和测试数据的比例和Batch size的大小，其中Noise ∈[0, 50]，Batch size ∈[1, 30]。

1. FEATURES（特征）：

FEATURES一栏包含了可供选择的7种特征：X1、X2、X12、X22、X1X2、sin(X1)、sin(X2)。X1可以看成以横坐标分布的数据特征，X2是以纵坐标分布的数据特征，X12和X22是非负的抛物线分布，X12是双曲抛物面分布，sin(X1)和sin(X2)是正弦分布。可通过这些特征的分布组合将两类数据（蓝色和黄色）区分开。

1. HIDDEN LAYERS（隐藏层）：

HIDDEN LAYERS一栏可设置多少隐含层。如何确定隐藏层的层数是一个至关重要的问题。首先需要注意，在神经网络中，当且仅当数据非线性分离时才需要隐藏层。理论而言，网络的层数越深，拟合函数的能力也越强，效果会更好，但是实际上更深的层数可能会带来过拟合的问题，同时也会增加训练难度，使模型难以收敛。因此需要根据当前面临的问题的特点与复杂性，合理的设置隐藏层数量。

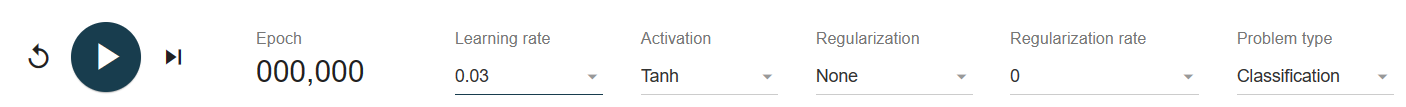
每个隐藏层可设置不同个数的neurons（神经元），如何确定隐藏层的神经元是一个至关重要的问题，在隐藏层中使用太少的神经元将导致欠拟合，使用过多的神经元可能会导致过拟合，最佳数量需要根据神经网络所解决的问题总结他人经验加上自己通过不断试验获得。

1. OUTPUT（输出层）：

OUTPUT一栏将输出的训练过程直接可视化，通过test loss和training loss来评估模型的好坏。一个好的网络，二者的差距应该是很低的。但一般情况下因为网络不可避免地存在一定程度上的过拟合，train loss会低于test lost，属于正常情况。但如果低太多，就得考虑是过拟合的问题还是因为样本的特征空间不统一的问题。

1. 其它参数：

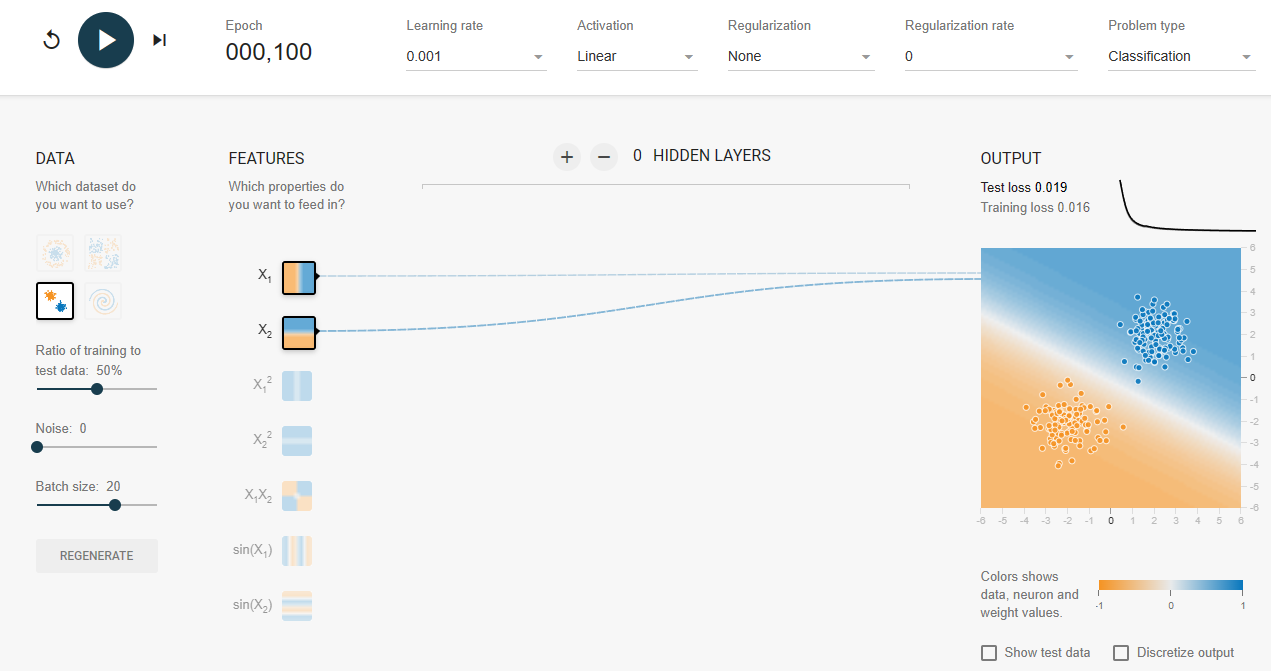
除了主要的四个部分外，在界面上还有一列控制神经网络的参数，从左到右分别是，训练的开关、迭代次数、学习速率、激活函数、正则化、正则化率和问题的类型。



1. 学习速率（Learning Rate）：学习速率是指在训练神经网络时，每次更新权重时所采用的步长大小。它决定了权重在每次迭代中的调整幅度。选择合适的学习速率对于模型的性能至关重要，过小的学习速率可能导致训练缓慢，而过大的学习速率可能导致训练不稳定。
2. 激活函数（Activation Function）：激活函数定义了神经网络中每个神经元的输出。它将输入值映射到非线性的输出，使神经网络能够学习和表示复杂的模式和关系。项目所提供的激活函数包括ReLU、Tanh、Sigmoid和Linear，每个激活函数都有其独特的性质和适用场景。
3. 正则化（Regularization）：正则化是一种用于防止过拟合的技术。过拟合是指模型过度适应训练数据，导致在未见过的数据上表现不佳。正则化通过在损失函数中引入额外的惩罚项，限制模型的复杂性，防止权重过大。项目可选择，None（无正则化）和L1、L2正则化。
4. 正则化率（Regularization Rate）：正则化率是用于控制正则化项对损失函数的贡献程度的超参数。较高的正则化率会对权重施加更强的惩罚，从而促使模型更加倾向于简化和泛化。
5. 问题的类型（Problem type）：项目仅提供分类问题（Classification）和回归问题（Regression）的建模，分类问题旨在将输入样本分为不同的类别，回归问题旨在预测连续的数值输出，可以根据特定问题的要求进行适当的架构和设置。
6. **试用分析**

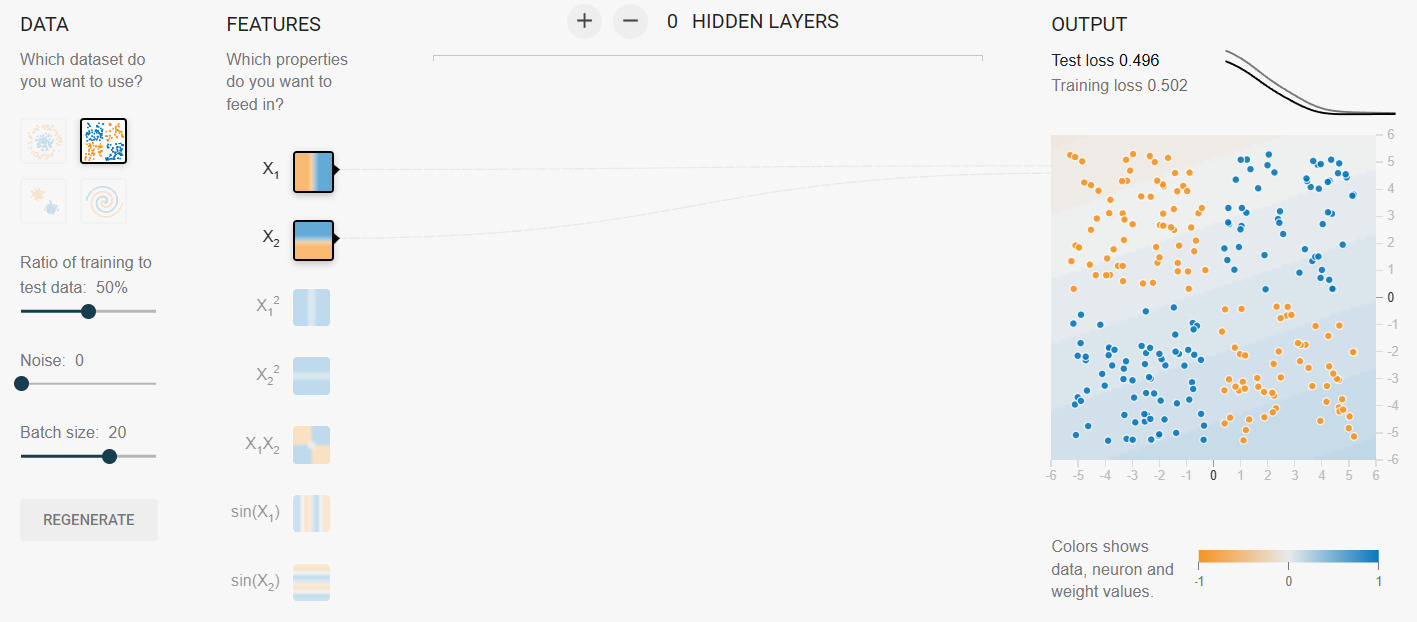
只使用线性特征+线性激活函数（Linear）处理分类问题，特征提取选择X1横坐标分布的数据特征和X2纵坐标分布的数据特征，设置学习率为0.001，不添加隐藏层，不采用正则化，此时模型只能处理简单的线性分布的数据，表现为使用一条直线对数据进行分割。对不同的分布数据经行训练。

处理高斯分布数据如下：

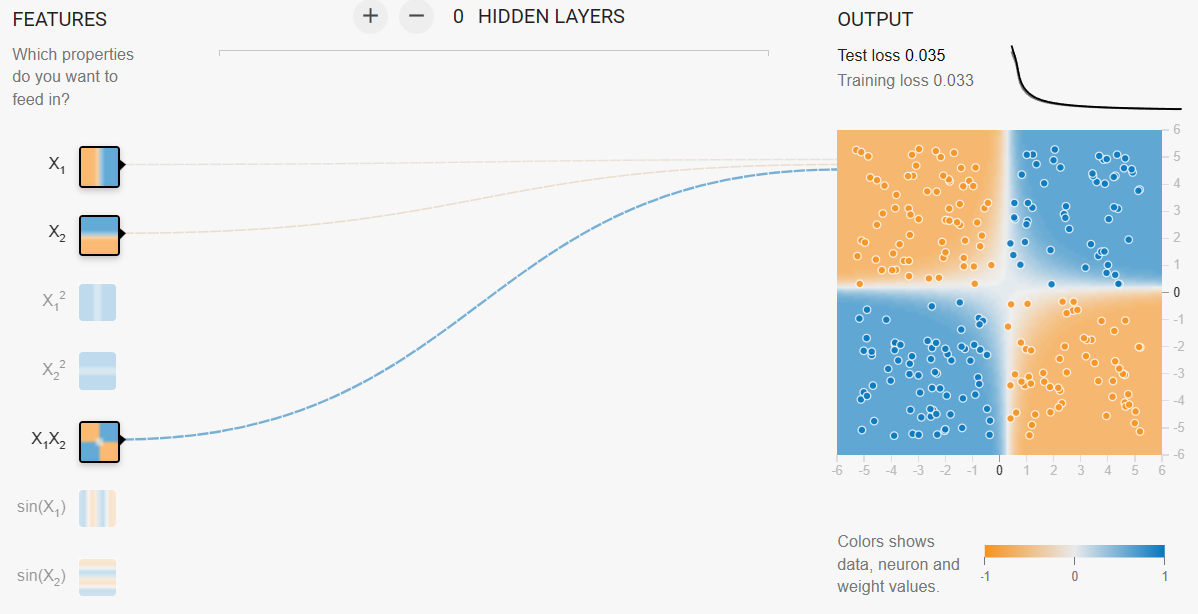


可以看到，经过100epoch后，Test loss = 0.019， Training loss = 0.016。满足Training loss < Test loss。且差距很小，观察loss 曲线发现两者的下降趋势一致，且基本重合，说明模型训练很成功，没有过拟合和欠拟合现象。

处理异或分布数据如下：

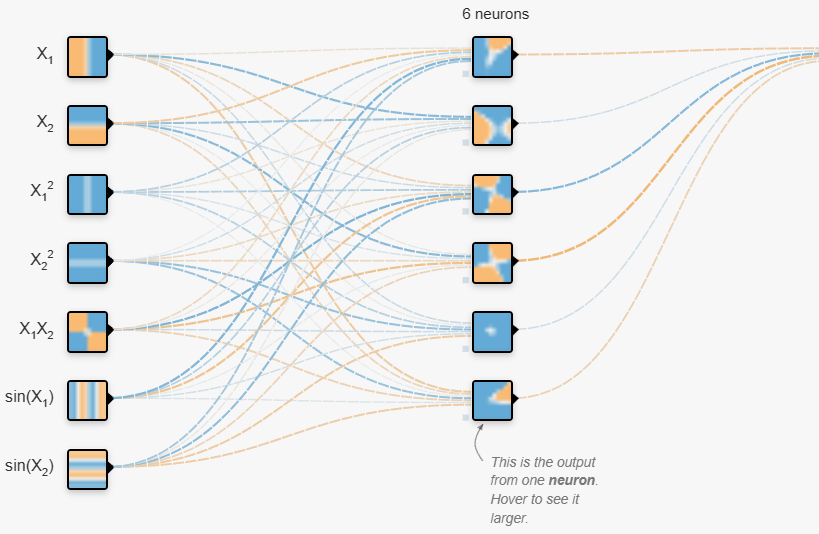


经过100epoch后，Test loss = 0.496，Training loss = 0.502，两者的损失值很大，说明该模型根本无法有效拟合。解决方法是引入X1X2 特征，重新训练100epoch，结果如下：

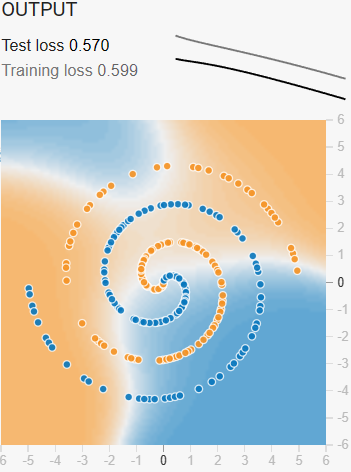
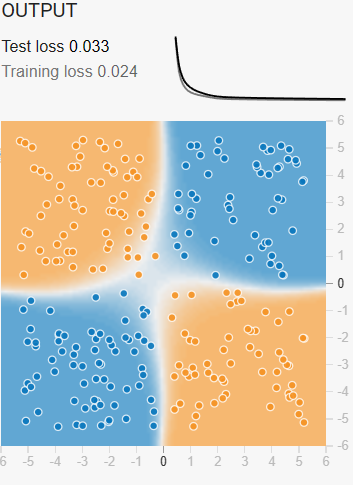
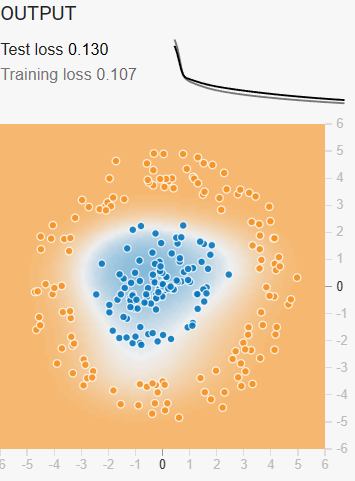


经过100epoch后，Test loss = 0.035，Training loss = 0.033，loss 曲线的下降趋势一致，且基本重合，可以看见，X1X2的特征的权重较高（来自X1X2的曲线较粗）。说明引入了X1X2后，模型已能够很好的拟合数据。

在上述模型的基础上，不对其它参数做修改，添加一层有6个神经元隐藏层，将所有的特征引入，构造一个全连接神经网络如下：

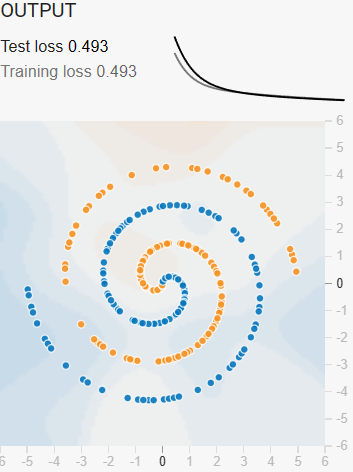
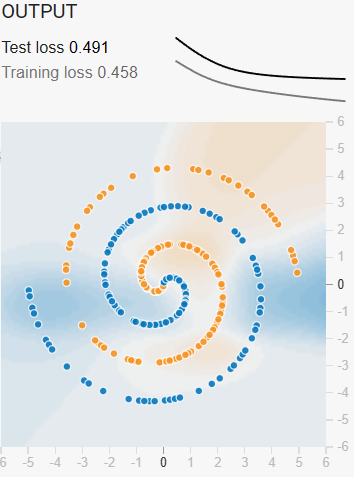
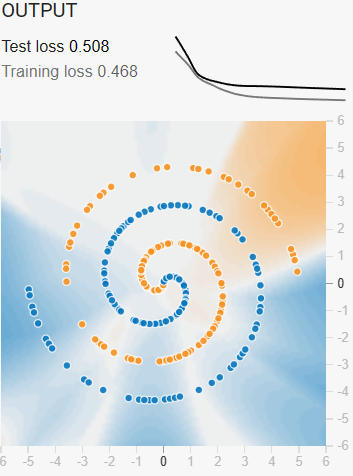


再分别尝试拟合圆形、异或和螺旋数据，100epoch的结果如下：

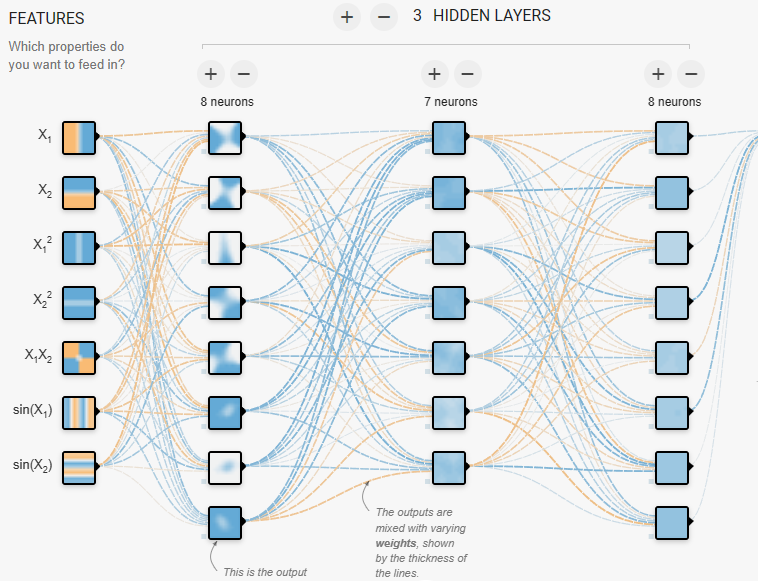


可以看见，模型对前两种类型的数据拟合情况较好，但对于最后的螺旋类型数据拟合情况较差。原因是由于模型只使用了线性激活函数（Linear），即使在特征层面引入了高斯和非线性，模型层面依然没有高斯和非线性，导致对最复杂的螺旋类型数据拟合较差。

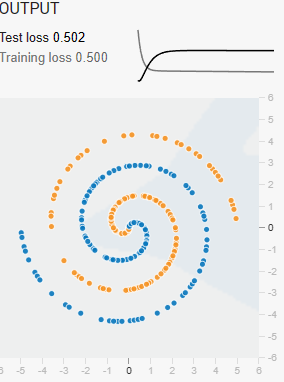
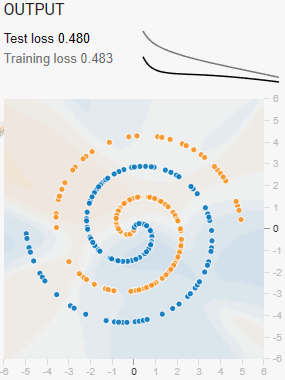
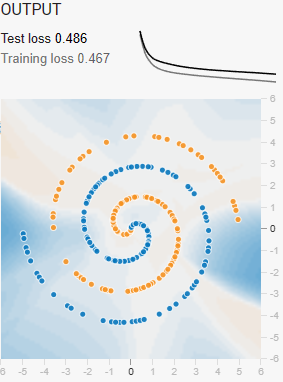
尝试将激活函数切换为其它的非线性激活函数，以下结果分别为 ReLU、Tanh和Sigmoid，对螺旋形数据拟合200epoch的结果：



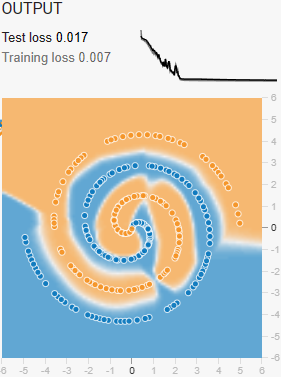
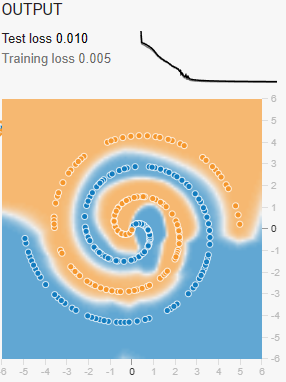
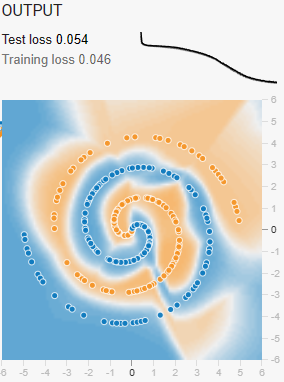
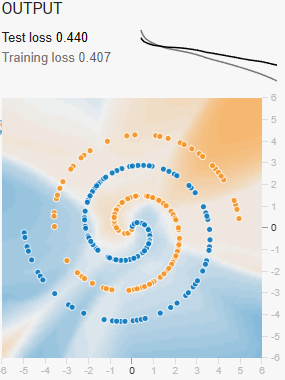
虽然引入了非线性激活函数后，数据权重分布（背景颜色的分布情况）不再单一，但数据拟合的情况依然较差。这里猜想的原因可能是模型太简单导致，尝试添加更多的隐藏层和神经单元，把模型结构变得更复杂，如下图所示：



以下结果分别为 ReLU、Tanh和Sigmoid，对模型训练200epoch的结果：



拟合的情况依然没有变得更好，原因是学习率设置得过低，导致数据收敛过慢，Sigmoid甚至出现了梯度消失的情况。这里只选用ReLU激活函数，增大学习率分别为0.003、0.01、0.03、0.1再次训练模型200epoch的结果如下。



学习率设置为0.003时，拟合速度过慢；学习率设置为0.01时，训练的模型已经能够比较好的拟合数据；学习率设置为0.03时，模型拟合效果最好，且最符合数据的特征；学习率设置为0.1时，前期的训练过程损失值出现了振荡，拟合效果并不如前一个设置的学习率模型好。至此，已经构建了一个合适的模型，并设置了合适的参数，使得该模型能够较好的拟合螺旋数据。