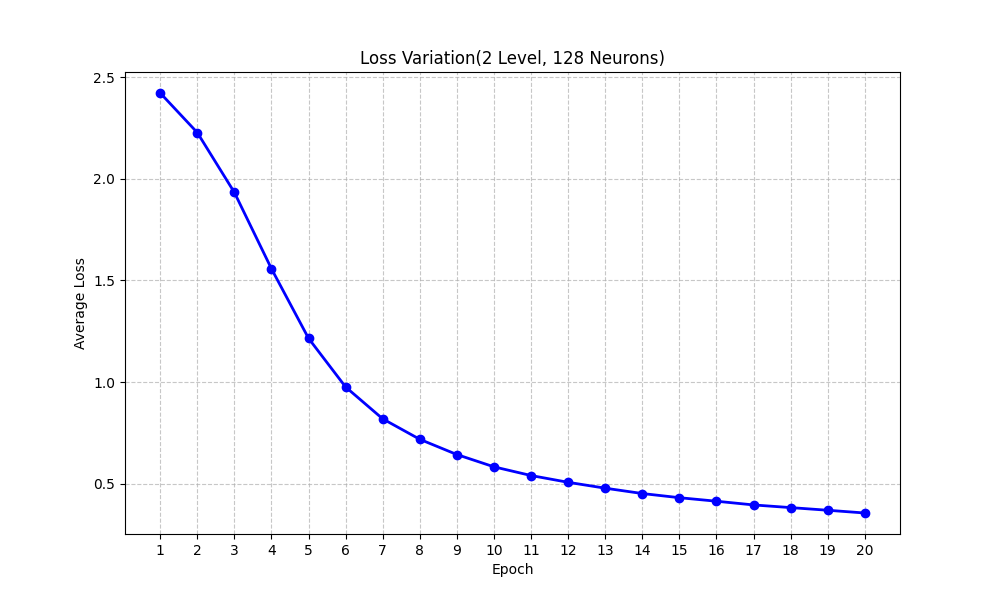
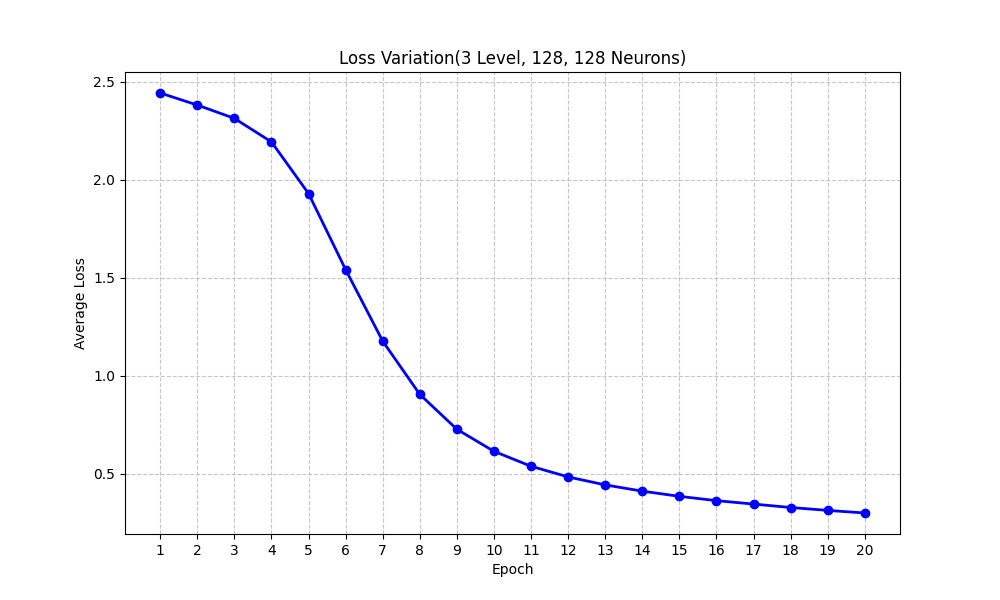
###### 作业1

采用了三层的神经网络，隐藏层每层128节点的方法训练。

首先将数据及整体迭代二十次，两层，隐藏层128节点和三层，隐藏层每层128节点得到的平均损失值变化如下两张图所示。

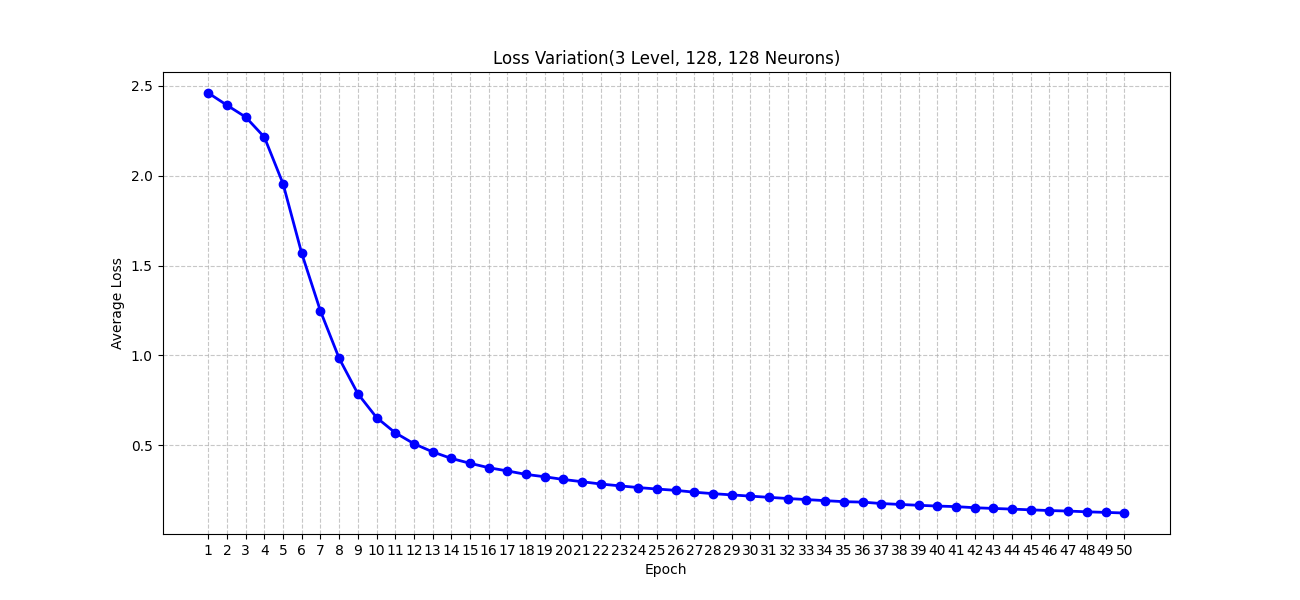
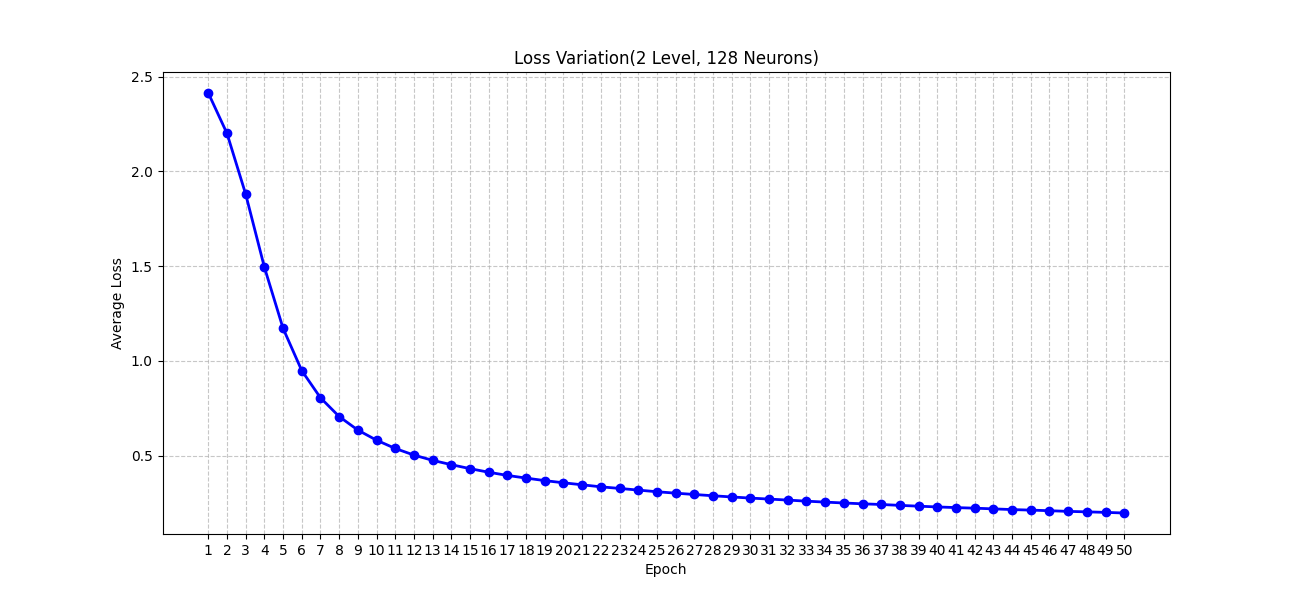
其实效果是差不多的，最后都收敛到0.25左右，但是第二个网络的损失值更低一些，拟合效果要稍微好于第一个网络





之后增加迭代的次数，增加到50次，趋势如下两张图所示。

损失值都收敛到大概1.9，效果没有明显的区别，拟合的效果都不错。

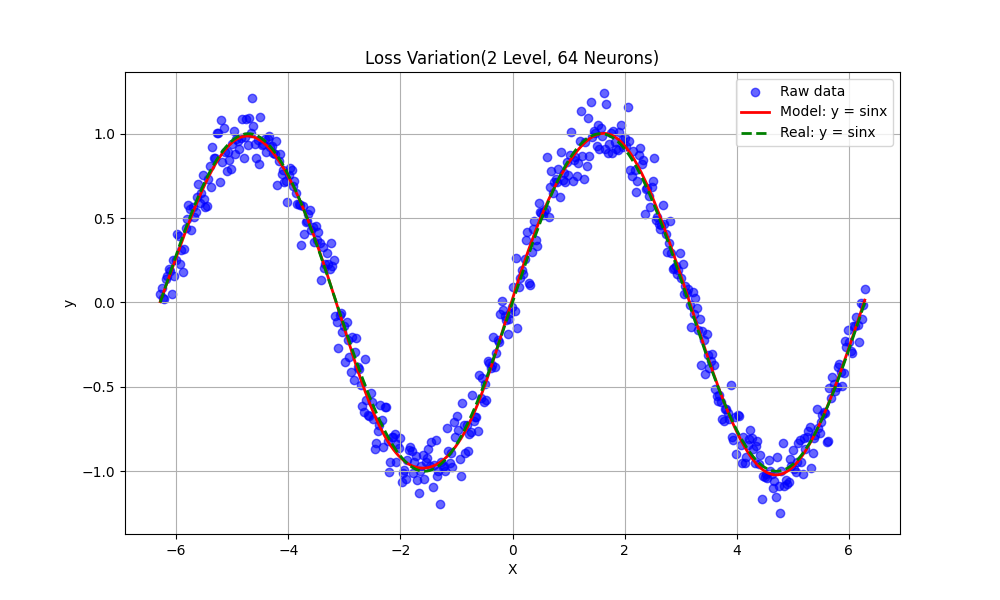


结论：可以看到这份数据2层的模型已经能够很好的拟合，得到一个不错的结果，如果要增加层数，效果没有明显提升并且还会降低效率，收敛会变慢，所以也不需要更加复杂的模型。

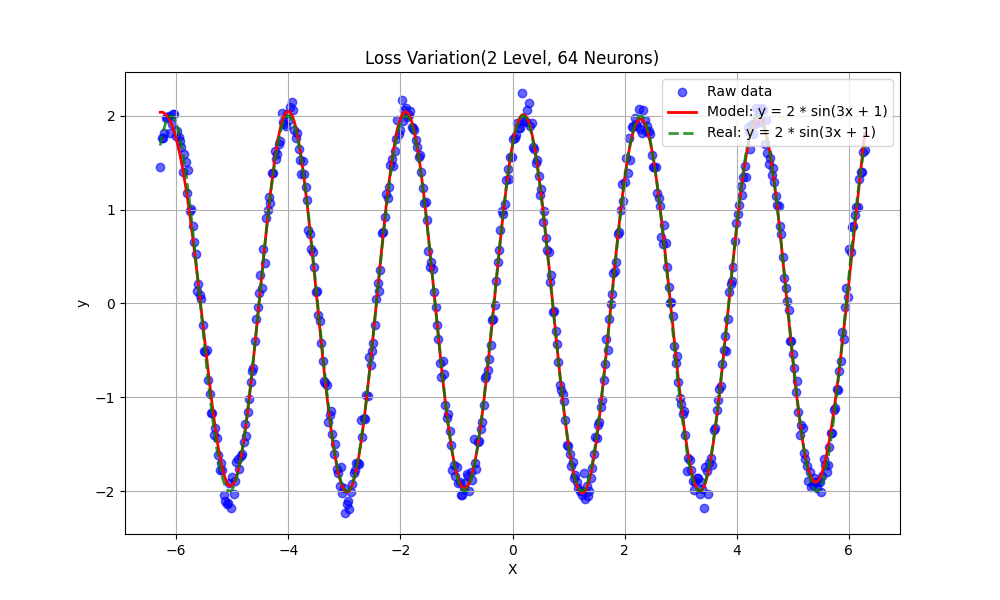
###### 作业2

首先拟合最简单的y = sinx，用2层，隐藏层64节点的神经网络，得到的结果如下图。

蓝色是生成的点，红色拟合的结果，绿色的虚线为真实的走向。可以看出，已经能够很好的拟合出最简单的正弦函数了。

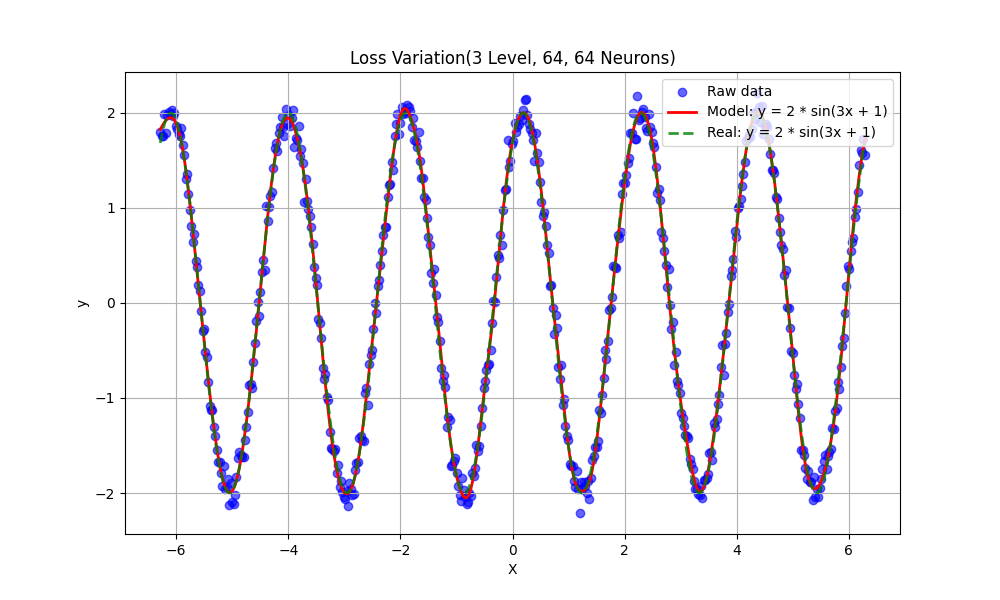


那如果是比较复杂的正弦函数呢？假设拟合的是y = 2sin(3x + 1)，用同样的神经网络来训练，得到的拟合结果如下图所示：



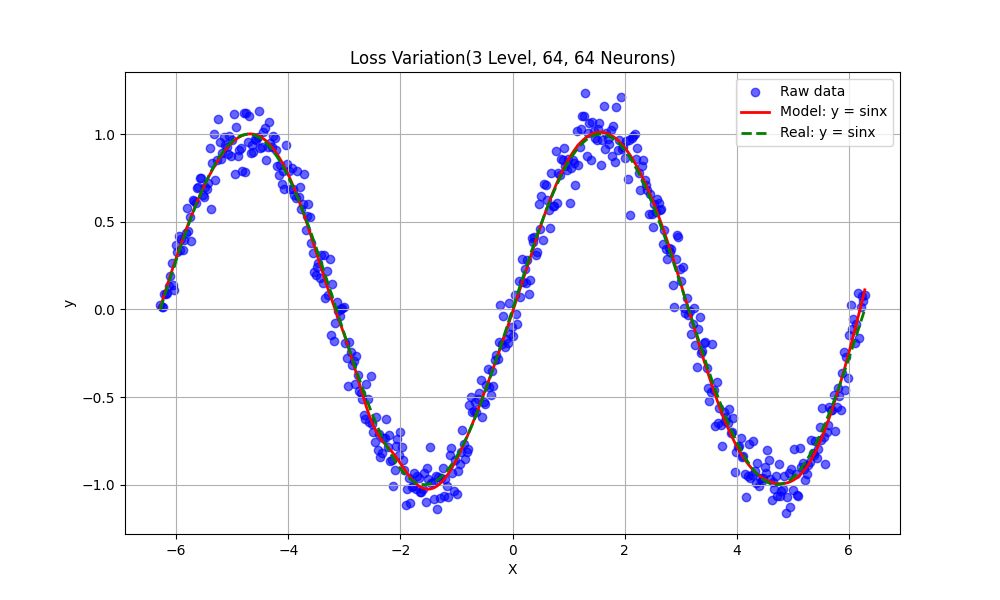
可以看出，拟合的效果也不错，但是在最左侧那里，拟合的不好，有一些偏移，和真实走向差别非常大。

所以尝试增加隐藏层数来拟合，用三层，隐藏层每层64个节点的神经网络来训练，得到的结果如下图所示：



可以看出，拟合效果相比两层的有了很大的改善，左侧不会出现偏差的情况出现。

同样，用这个模型拟合y = sinx，结果如下图所示：



可以看到拟合效果也非常好，不过和两层的神经网络差别不大。所以对于比较简单的函数，可以用两层的网络，效果好而且效率高，但是对于复杂的函数，就需要增加层数和节点来拟合，效果会有很大的改善。