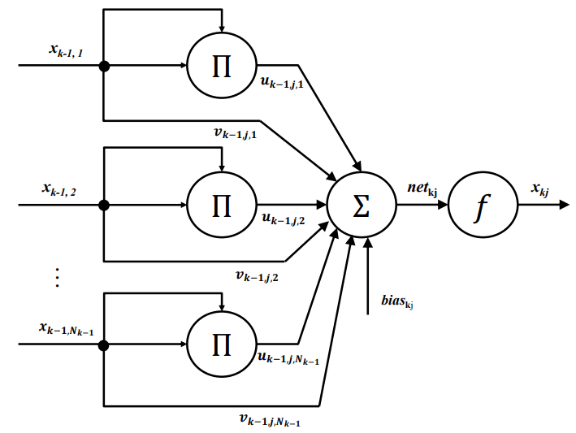
**Assignment #1**

**Problem 1：**

 **对于一个MLQP网络中的计算单元进行反向梯度计算推导：**

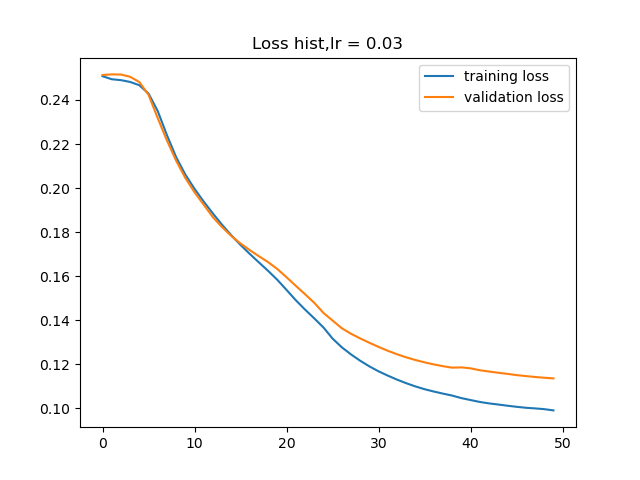
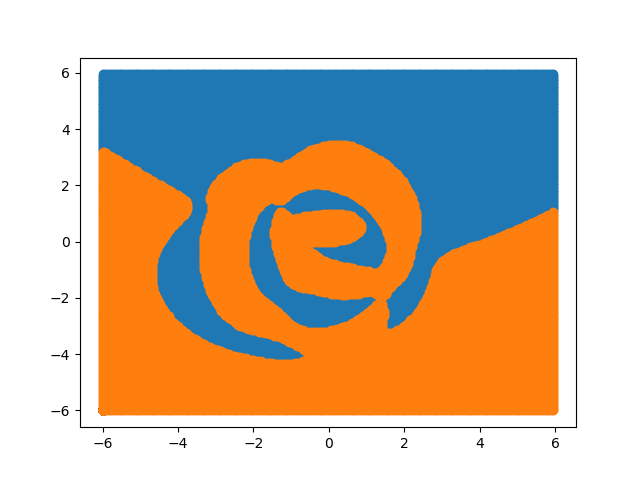
**正向传播计算公式：，其中是Sigmoid激活函数**

**则根据链式法则求反向传播公式：**

**Problem 2：**

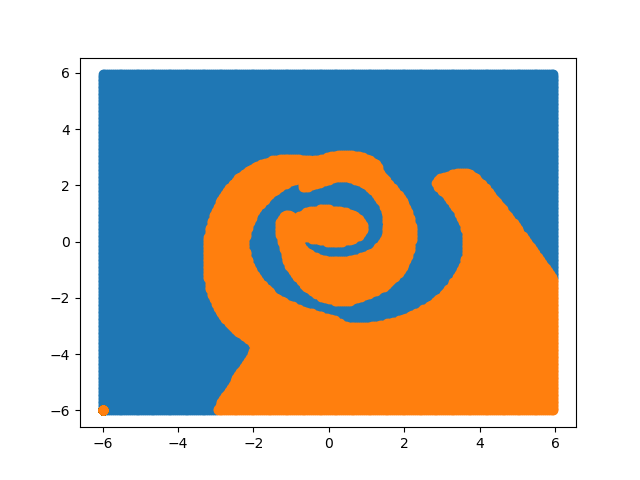
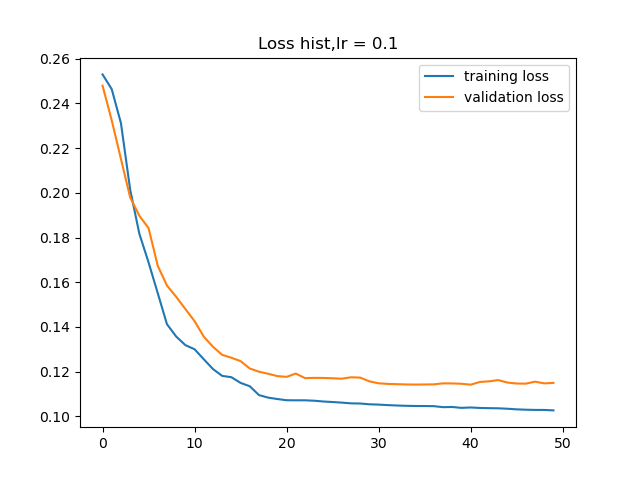
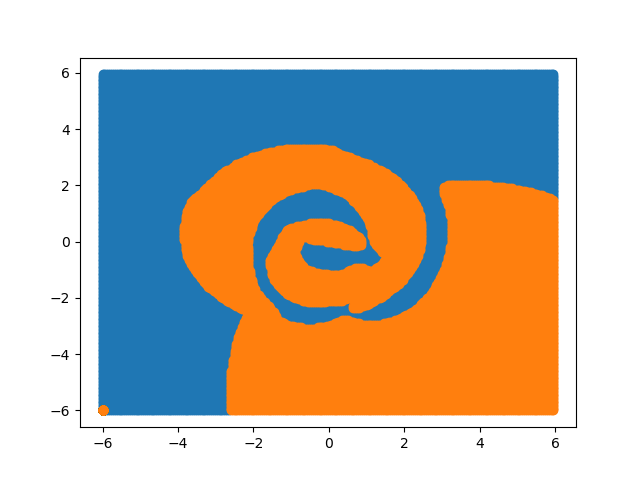
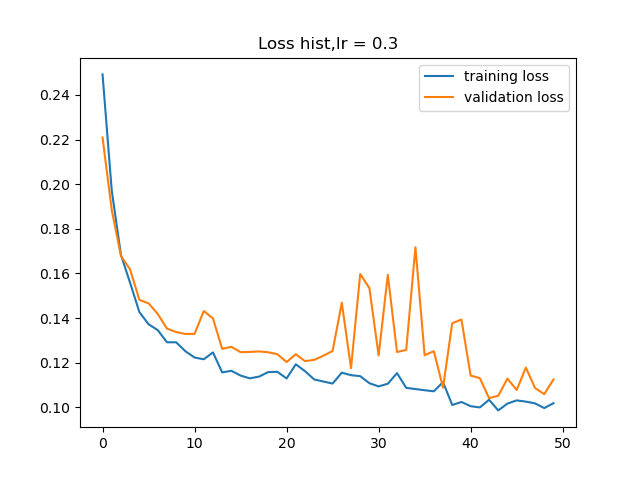
**本项目使用了pytorch进行仿真。另外，为防止原始数据集样本个数不足，自行编写了样本生成函数，生成了更多样本进行训练。**

**在使用MLQP网络的基础上添加了单层隐藏层，节点数为16。输出层为单节点，在使用sigmoid函数激活后施加硬判决，损失函数采用MSE。三种不同的学习率下的仿真结果如图所示。**



**Fig.1 lr = 0.03**

**Fig.2 lr = 0.1**



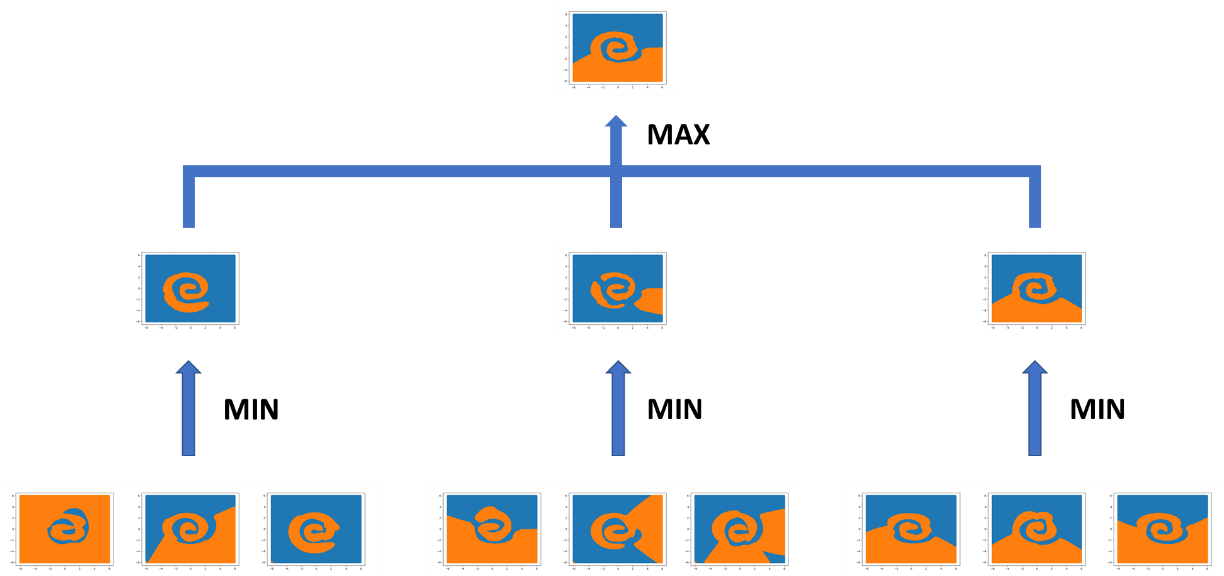
**Fig.3 lr = 0.3**

**结果分析：**

**首先可以看出，使用三种不同学习率进行训练时，尽管速度存在快慢只差，但最后在训练集中loss都能大概收敛至0.10，最后画出的判决边界在准确度上也大致相同。**

**然而，不同大小的学习率会导致不同的收敛速度。当lr = 0.03时，收敛速度较慢，当epoch = 20 左右出现过拟合现象，epoch = 45 左右开始逐渐收敛。当lr = 0.1时，收敛速度适当，当epoch = 8左右出现过拟合现象，epoch = 20 左右就开始收敛了。而 lr = 0.3对于这个项目而言显然过大了，在 epoch = 5时出现过拟合现象，并且在训练后期验证集的loss出现震荡现象，无法稳定收敛。**

**Problem 3：**

** 将label = 0以及label = 1的训练数据各分为3份，组合之后构成9份训练集分别训练，将训练结果采用MIN-MAX的方式进行组合，结果如图所示。**

**结果分析：**

**可以看出，在本次实验中使用MIN-MAX方式将子问题结果进行组合后，模型效果并没有得到显著提升。推测结果可能是自主产生样本数量过多，而模型本身架构又过于简单，导致仅仅训练子问题的数据集就达到了模型精度的极限。在这基础上使用MIN-MAX方案并不会带来更进一步的增益。**