

Task 1

代码基本架构

- layer. py : 代表的是神经网络的层, 包含了该层**线性变化**和**非线性变化** (可以设置为没有)
- network. py : 根据输入的参数构建的神经网络, 包含backward forward方法
- sin
 - 文件夹都是数据
 - train_network_sin.py : 训练神经网络的程序
 - sin_data_model.py : 数据加载器
 - experiencescript. py : 实验脚本代码
 - trained_network.py : 加载训练好的模型
- img
 - 文件夹都是数据
 - dataloader. py
 - imgdataset. py 以上为模仿pytorch 的dataset 和dataloader写的数据集处理机制
 - train_network_img.py : 训练神经网络的程序
 - experiencescript. py : 实验脚本代码
 - trained_network=.py : 加载训练好的模型

实验

拟合 Sin

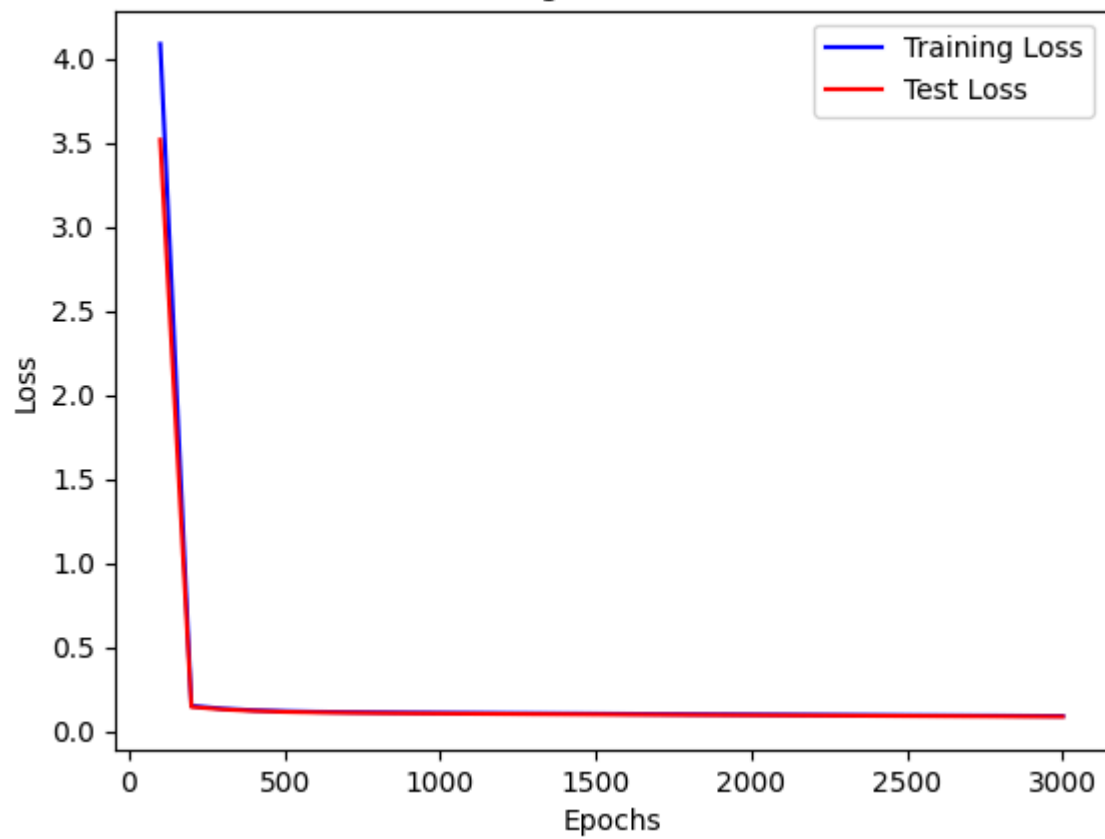
学习率对网络的影响

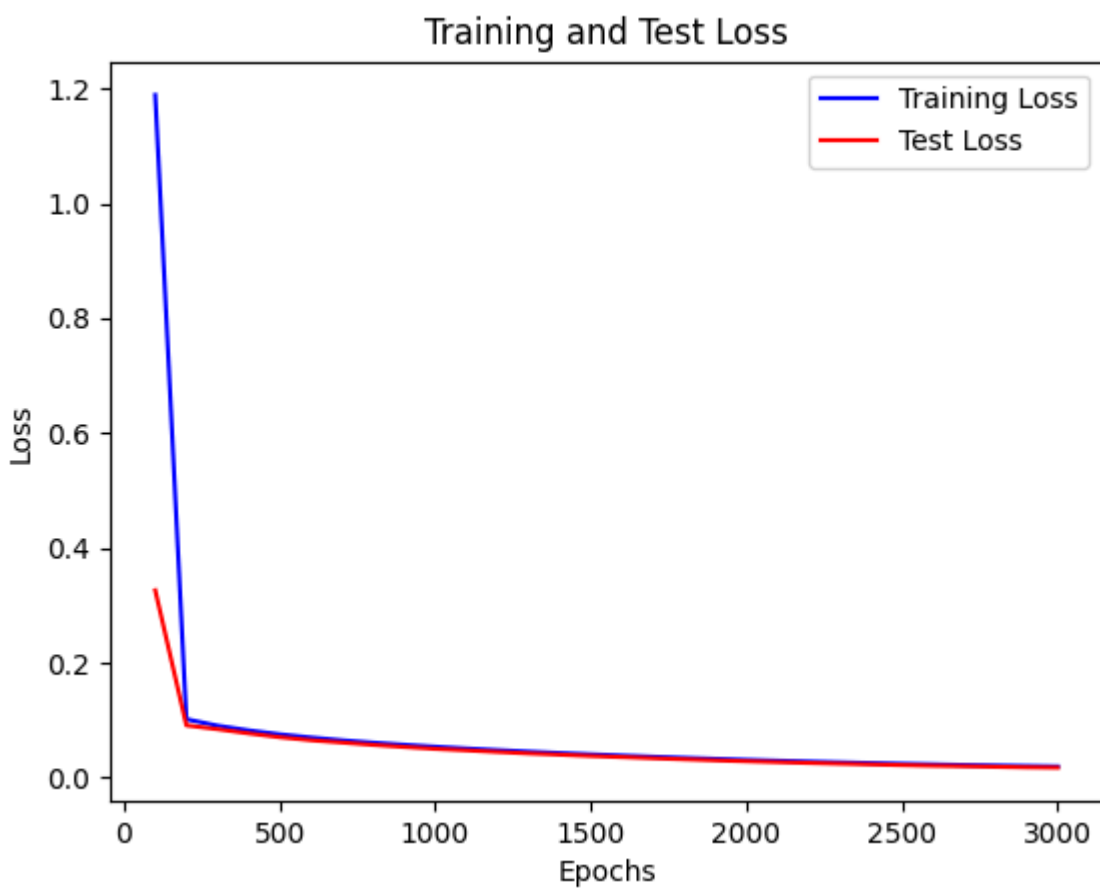
- 基于 单层 32unit sigmoid激活函数 实验
- 上图为lr为0.001 下图为 0.01 的情况

可见 对于sigmoid函数而言

lr太小的话, 在部分epoch之后, loss会开始慢速下降, 导致训练缓慢, 所以需要较大的lr

Training and Test Loss

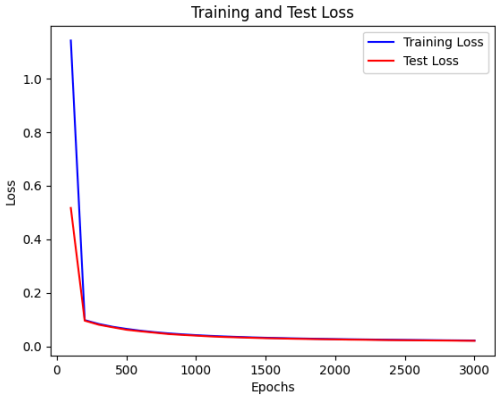




正则项对网络的影响

- 下面的两个网络在相同初始化情况下进行迭代
- 可见 正则项会惩罚过大的梯度，导致下降较慢，同时会防止过拟合

网络	L1误差	图像
正则项系数为0	Epoch 2900, Loss: 0.0208701047369986	

网络	L1误差	图像
正则项系数为0.01	Epoch 2900, Loss: 0.022429724348479462	

网络层数对网络的影响

- 实验：对两个不同网络同时进行十次测试 得到结果 取平均
- 可见单层网络收敛速度较快
- 双层网络更容易在不改变学习率的情况下不收敛

网络	3000epoch后的L1误差
sigmoid 单层网络	0.014551067166471124
sigmoid双层网络	0.02297293823671869

单层网络的神经元个数对网络的影响

- 实验：对两个不同单层网络同时进行十次测试 得到结果 取平均
- 可见在拟合任务上 单层情况下 越多的神经元数量会使得拟合的更快 且更准确
- 由于是拟合问题 所以神经元过多导致的过拟合问题就不会显现

网络	3000epoch后的L1误差
32个神经元	0.014551067166471124
128个神经元	0.006307620398259217

图像分类

激活函数对网路的影响

学习率对网络的影响

正则项对网络的影响

网络层数对网络的影响

单层网络的神经元个数对网络的影响

对反向传播算法的理解