

DL lab1

吴越凡 PB18000149

April 2021

1 实验目的

利用 pytorch 或者 tensorflow 去实现前馈神经网络去拟合 $\sin x$ 函数。同时研究学习率，网络宽度，网络深度，激活函数等对性能的影响。

2 算法描述

```
class DNN(nn.Module):
    def __init__(self, hid_dim=4, depth=4):
        super(DNN, self).__init__()
        self.linears = nn.ModuleList()
        self.linears.append(nn.Linear(1, hid_dim))
        self.linears.append(nn.Tanh())
        for i in range(depth-2):
            self.linears.append(nn.Linear(hid_dim, hid_dim))
            self.linears.append(nn.Tanh())
        self.linears.append(nn.Linear(hid_dim, 1))
    def forward(self, x):
        for layer in self.linears:
            x = layer(x)
        return x
```

DNN 是一个基于父类 `nn.Module` 的子类，参数读入 hidden layer 的神经元数量，使用 forward 连接。

对于学习率，仅选择了 0.01 和 0.001。

激活函数只选择了几个常用的，比如 \tanh 、 sigmoid 、 gelu 。其中常见的 relu 激活函数，由于其仅在正半轴有值，不符合 $\sin x$ 函数的情况，故将其删去。

对于深度，仅考虑了 4, 5, 6，三种情况。其隐藏层的层数分别为 2, 3, 4。

而宽度，仅选择了 8, 16, 32, 64 这四种情况。

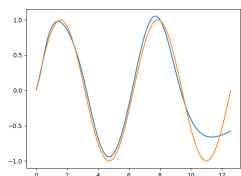
3 实验结果

从 0 到 4π 间采样 1000 个点，输入网络得到结果去得到对应的值，连接成曲线。具体的各参数值的结果在下方有汇总。

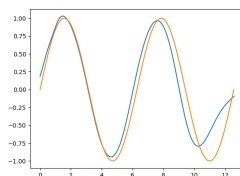
从下面图中的结果中可以看出，拟合的最好的是取宽度为 32，深度为 5，激活函数为 \tanh ，学习率为 0.001 的情况。其在大部分的区域上近乎完全重合。

4 总结

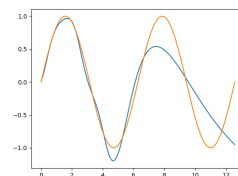
- 从总体上来看，对于网络宽度为 8 的时候，拟合的性能相对来说没有很好，随着宽度的增加，性能慢慢提高，直到宽度为 64 的时候，出现了个别偏差较大的情况。
- 从实验结果来讲，激活函数最佳的选择是 \tanh 。一部分可能是 \tanh 存在关于 0 对称的两部分，并且其较为平滑。
- 深度对于实验结果的影响，单从效果上来看并没有特别大。对应于每一种深度，都存在拟合效果较好和拟合效果较差的情况。可能是由于 $\sin x$ 函数较为简单，对于深度的需求没有那么大。



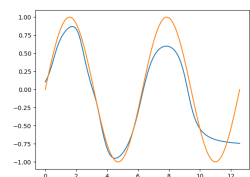
(1) Wid16-depth4-gelu-0.001



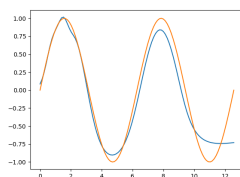
(2) Wid16-depth4-gelu-0.01



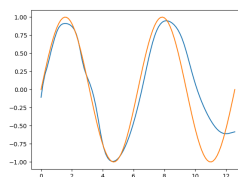
(3) Wid16-depth4-sigmoid-0.001



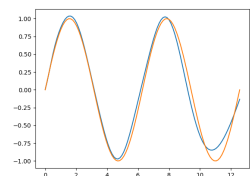
(4) Wid16-depth4-sigmoid-0.01



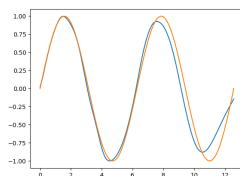
(5) Wid16-depth4-tanh-0.001



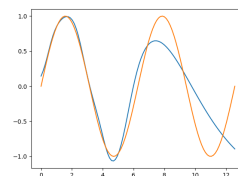
(6) Wid16-depth4-tanh-0.01



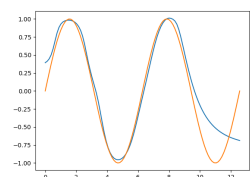
(7) Wid16-depth5-gelu-0.001



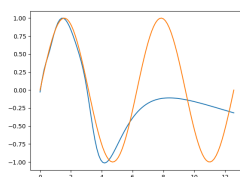
(8) Wid16-depth5-gelu-0.01



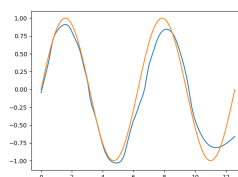
(9) Wid16-depth5-sigmoid-0.001



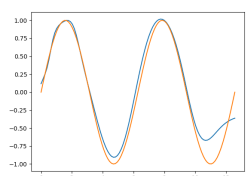
(10) Wid16-depth5-sigmoid-0.01



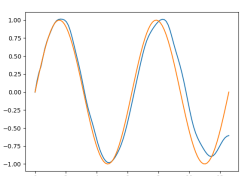
(11) Wid16-depth5-tanh-0.001



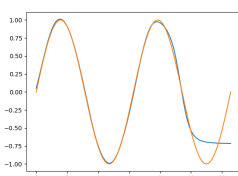
(12) Wid16-depth5-tanh-0.01



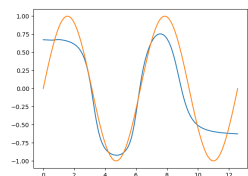
(13) Wid16-depth6-gelu-0.001



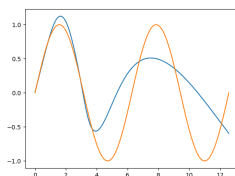
(14) Wid16-depth6-gelu-0.01



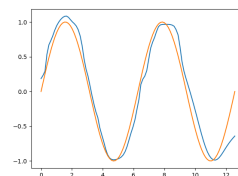
(15) Wid16-depth6-sigmoid-0.001



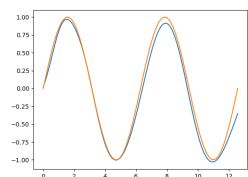
(16) Wid16-depth6-sigmoid-0.01



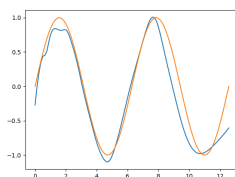
(17) Wid16-depth6-tanh-0.001



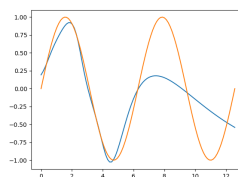
(18) Wid16-depth6-tanh-0.01



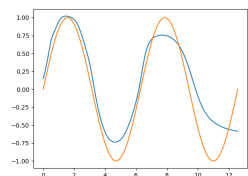
(19) Wid32-depth4-gelu-0.001



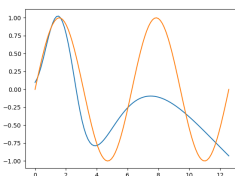
(20) Wid32-depth4-gelu-0.01



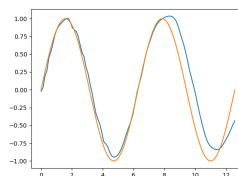
(21) Wid32-depth4-sigmoid-0.001



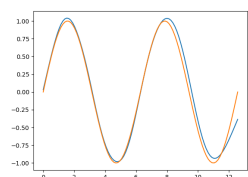
(22) Wid32-depth4-sigmoid-0.01



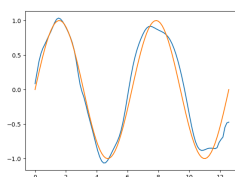
(23) Wid32-depth4-tanh-0.001



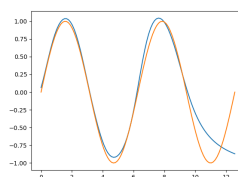
(24) Wid32-depth4-tanh-0.01



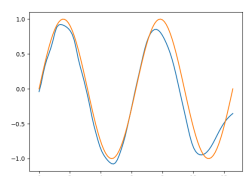
(25) Wid32-depth5-gelu-0.001



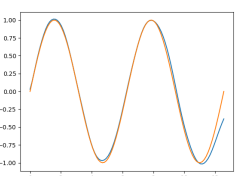
(26) Wid32-depth5-gelu-0.01



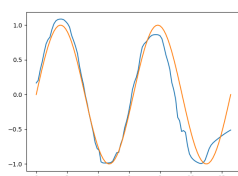
(27) Wid32-depth5-sigmoid-0.001



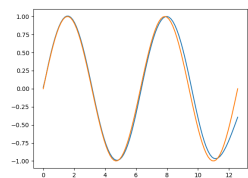
(28) Wid32-depth5-sigmoid-0.01



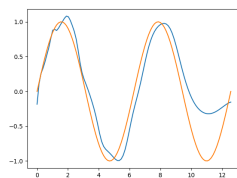
(29) Wid32-depth5-tanh-0.001



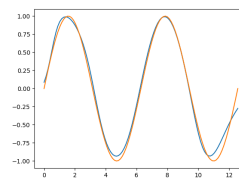
(30) Wid32-depth5-tanh-0.01



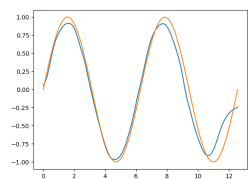
(31) Wid32-depth6-gelu-0.001



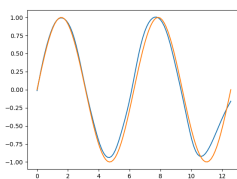
(32) Wid32-depth6-gelu-0.01



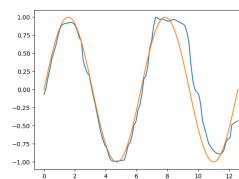
(33) Wid32-depth6-sigmoid-0.001



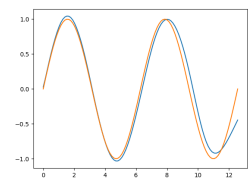
(34) Wid32-depth6-sigmoid-0.01



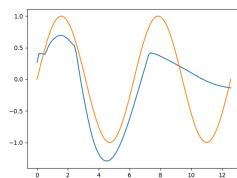
(35) Wid32-depth6-tanh-0.001



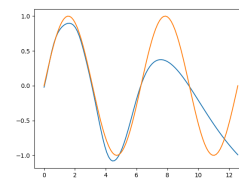
(36) Wid32-depth6-tanh-0.01



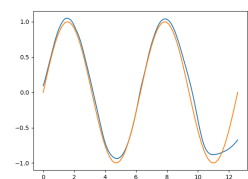
(37) Wid64-depth4-gelu-0.001



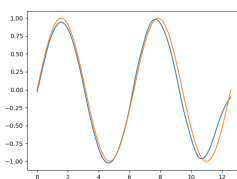
(38) Wid64-depth4-gelu-0.01



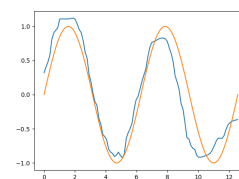
(39) Wid64-depth4-sigmoid-0.001



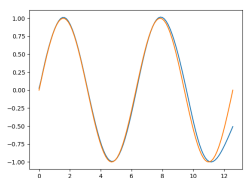
(40) Wid64-depth4-sigmoid-0.01



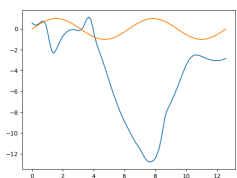
(41) Wid64-depth4-tanh-0.001



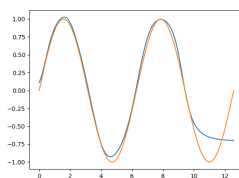
(42) Wid64-depth4-tanh-0.01



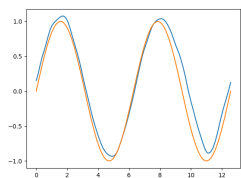
(43) Wid64-depth5-gelu-0.001



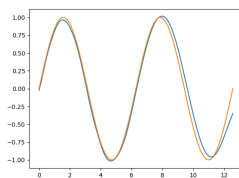
(44) Wid64-depth5-gelu-0.01



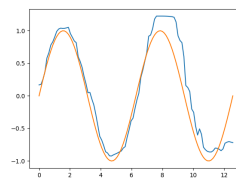
(45) Wid64-depth5-sigmoid-0.001



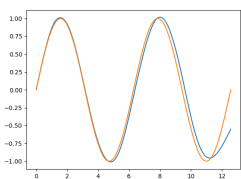
(46) Wid64-depth5-sigmoid-0.01



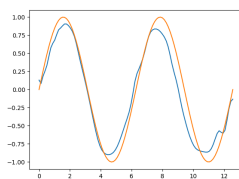
(47) Wid64-depth5-tanh-0.001



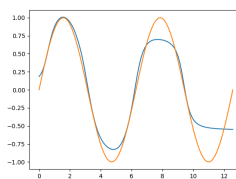
(48) Wid64-depth5-tanh-0.01



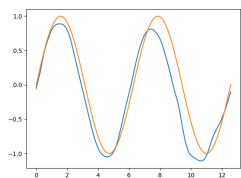
(49) Wid64-depth6-gelu-0.001



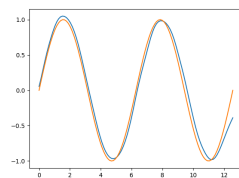
(50) Wid64-depth6-gelu-0.01



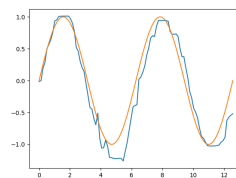
(51) Wid64-depth6-sigmoid-0.001



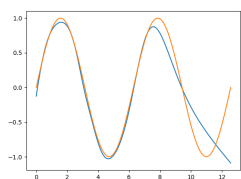
(52) Wid64-depth6-sigmoid-0.01



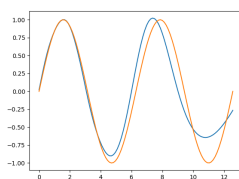
(53) Wid64-depth6-tanh-0.001



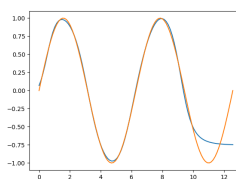
(54) Wid64-depth6-tanh-0.01



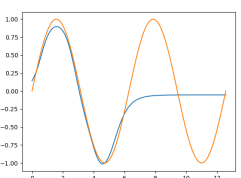
(55) Wid8-depth4-gelu-0.001



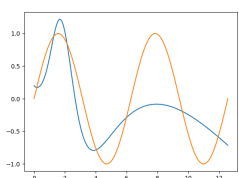
(56) Wid8-depth4-gelu-0.01



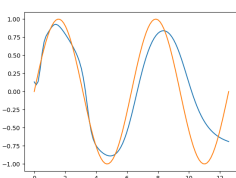
(57) Wid8-depth4-sigmoid-0.001



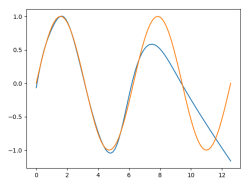
(58) Wid8-depth4-sigmoid-0.01



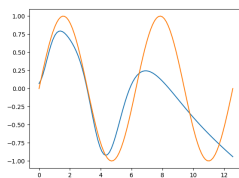
(59) Wid8-depth4-tanh-0.001



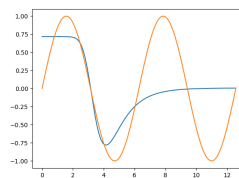
(60) Wid8-depth4-tanh-0.01



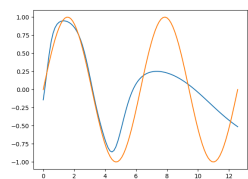
(61) Wid8-depth5-gelu-0.001



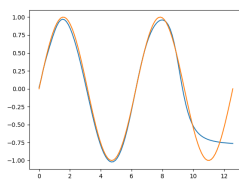
(62) Wid8-depth5-gelu-0.01



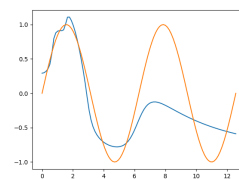
(63) Wid8-depth5-sigmoid-0.001



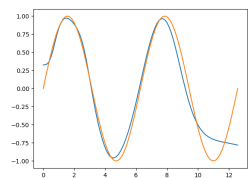
(64) Wid8-depth5-sigmoid-0.01



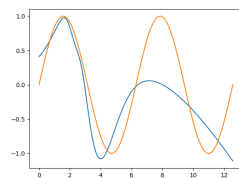
(65) Wid8-depth5-tanh-0.001



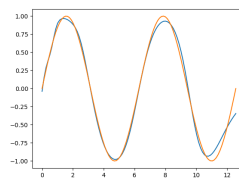
(66) Wid8-depth5-tanh-0.01



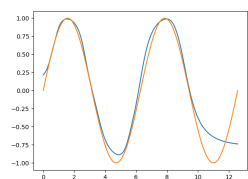
(67) Wid8-depth6-gelu-0.001



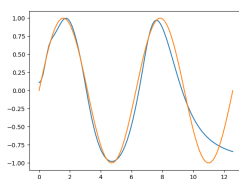
(68) Wid8-depth6-gelu-0.01



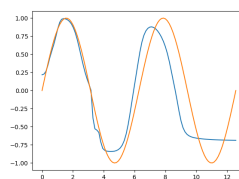
(69) Wid8-depth6-sigmoid-0.001



(70) Wid8-depth6-sigmoid-0.01



(71) Wid8-depth6-tanh-0.001



(72) Wid8-depth6-tanh-0.01