Assignment

佘国榛

15307130224

Proof

引理：如果一个连续函数不能被多项式表达，则此函数可以用来作为通用近似定理的推广条件。

下面证明引理成立，用反证法：

如果对于p(x) = Relu(x), 存在

p(x) = a_0 +a_1 * x + a_2 * x ^ 2 +a_3 * x ^ 4.....

当x < 0时， p(x) = 0恒成立，这表明该多项式方程有无数个根，与代数定理n次方程最多n个根矛盾，所以假设不成立，故由引理可知，双层Relu可以模拟任何连续函数

ru

Simulation

函数定义

a. y = x^2 b. y = e^x c. y = sinx d. y = lnx e. y = sinx + cosx

模型描述

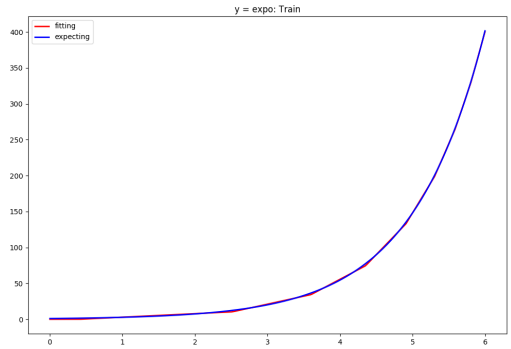
利用pytorch作为拟合所用的神经网络的框架，使用线性层加单层Relu作为基础，随即初始化函数，

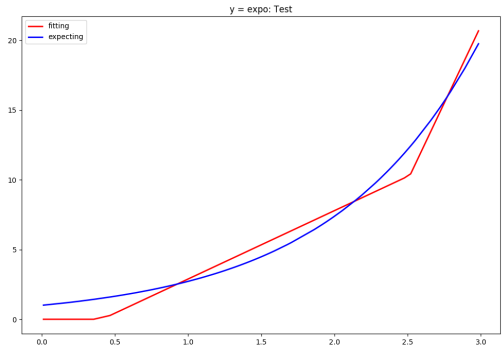
对于不同的神经网络，随机化初始值，调节epoch次数，神经元个数，就可以得到最后数据，网络的代码如下：

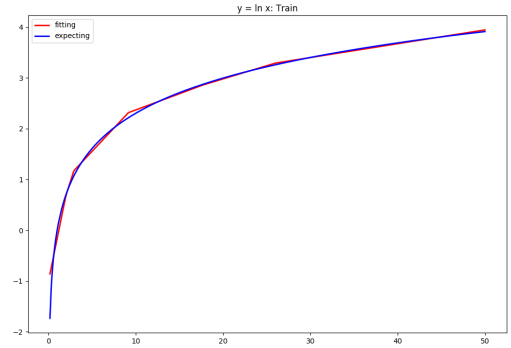
```
class DoubleReluNet(nn.Module):

    def __init__(self):
        super(DoubleReluNet, self).__init__()

        self.fc1 = nn.Linear(1, 20)
        # self.fc2 = nn.Linear(5, 5)
        self.fc3 = nn.Linear(20, 1)

    def forward(self, x):

        x = F.relu(self.fc1(x))
        # x = F.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```

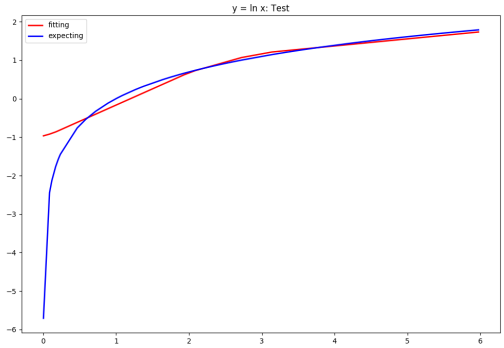
拟合结果

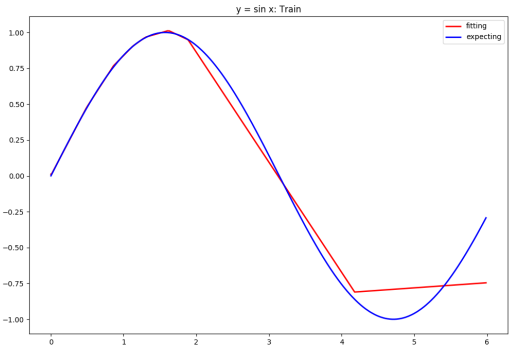
由于初始化过程的随机性， 我们并不是每次都可以得到最好的拟合结果，所以我们会将结果序列化保存在硬盘上，下面是不同函数的拟合结果。

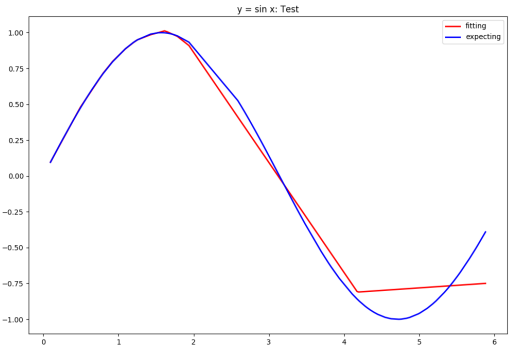


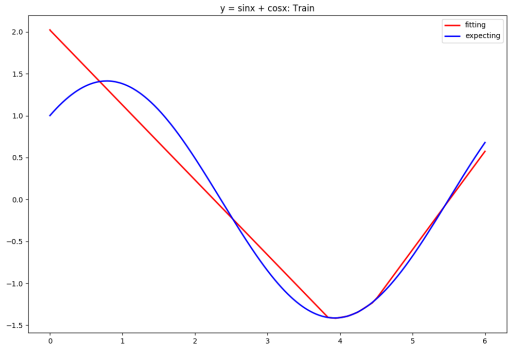


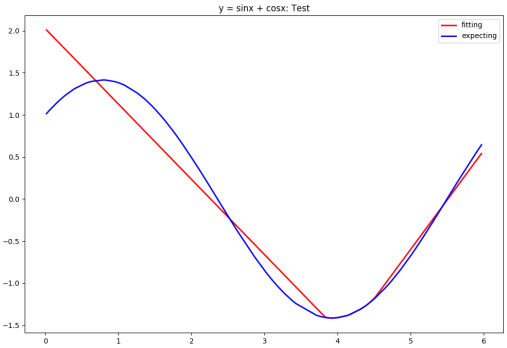


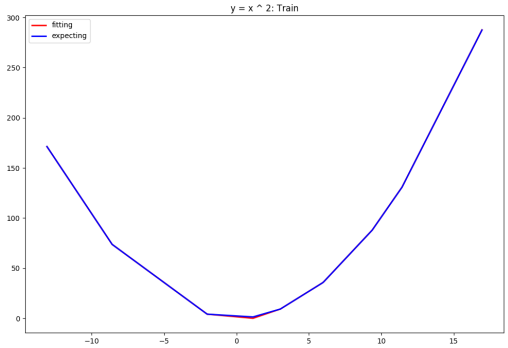


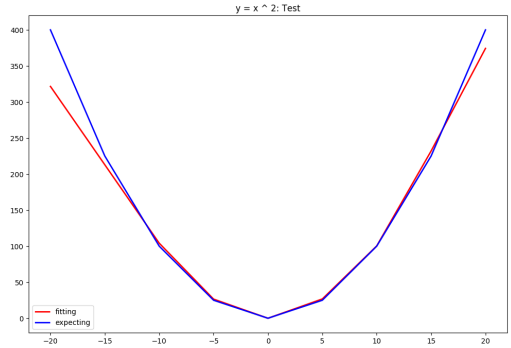












实验分析

这个实验在验证的重点是，如何调节神经网络的参数，在代码中针对了不同的拟合函数设置了不同的方法，最终达到了效果，在选取随机sampling的时候，我们需要针对函数的值规定范围。这当中我们发现，在训练sinx等非单调函数时，往往需要更多的迭代次数，才能达到相对较好的拟合效果，这和我们的预期一致，也说明了神经网络的泛化能力极强。