homework 2

作业二是构建RBF网络，来进行函数拟合。这次作业中，我使用pytorch来完成神经网络的构建，并且利用matplotlib来进行可视化。

RBF函数

此次使用的径向基函数为Gauss函数。Gauss函数标准形式如下：

$$g_{\mu,\sigma}(x)=\frac{1}{\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

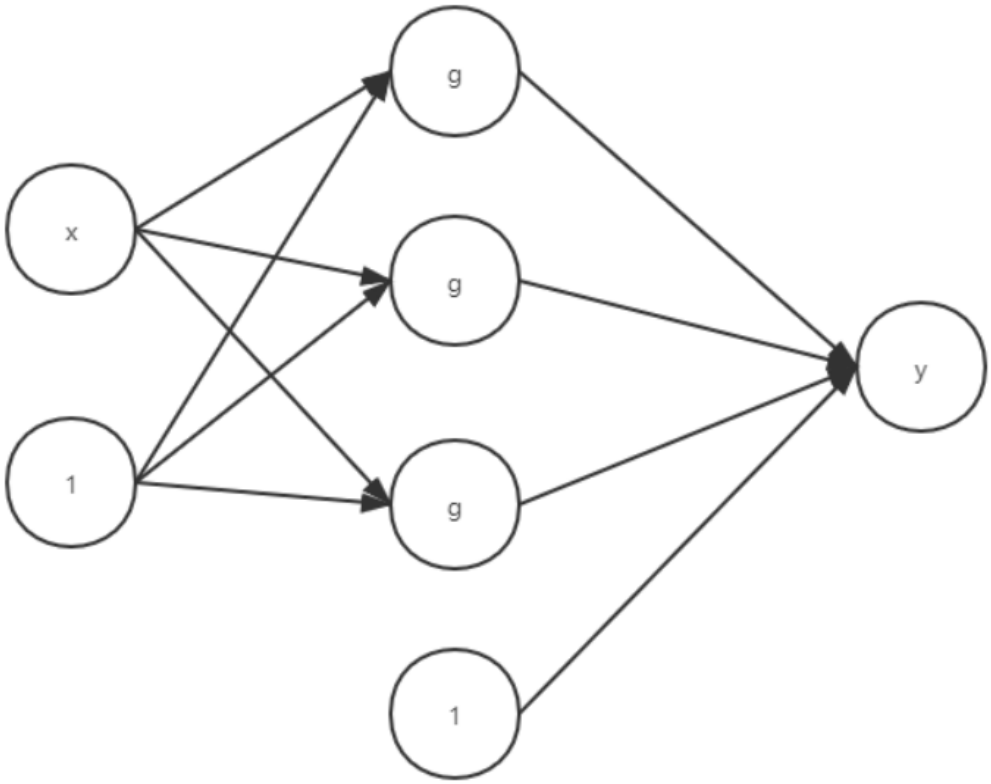
通过将形式组织成下面的函数形式：

$$g_{\mu,\sigma}(x)=\frac{1}{\sqrt{2\pi}}e^{-\frac{(x-\mu)^2}{2\sigma^2}}=\frac{1}{\sqrt{2\pi}}e^{-\frac{1}{2}\left(\frac{x}{\sigma}-\frac{\mu}{\sigma}\right)^2}=g_{0,1}(ax+b)$$

我们将原来高斯函数中的μ,σ变成了可以学习的线性参数a,b，可以轻易通过torch中的一个线性层来实现。

网络结构

我实现的RBF网络结构非常简单，只有一个隐藏层，如下图：



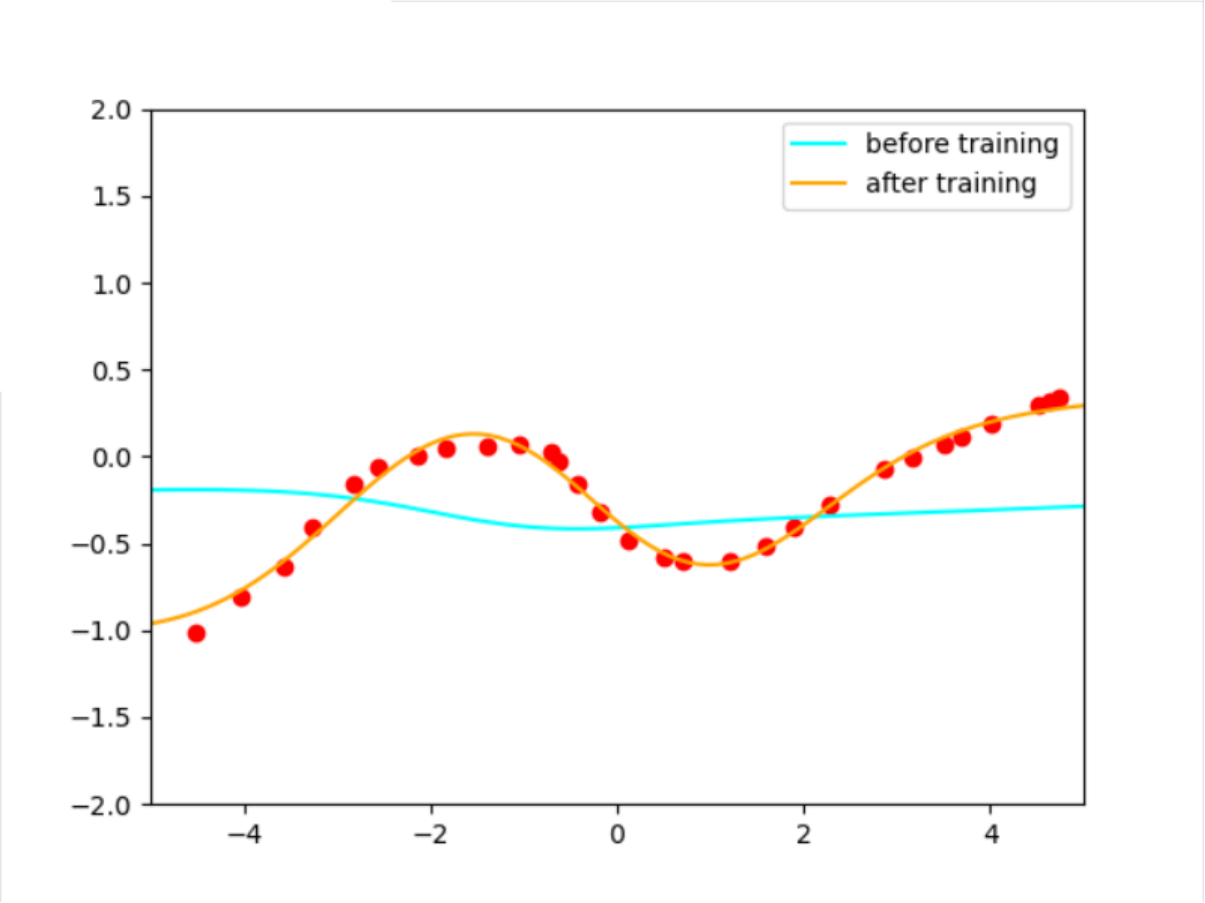
Hidden层中神经元的个数n，代表了函数空间的自由度。在函数插值时，一般需要自由度等于点的个数才能保证得到精确解，在神经网络中，n可以自由设定。在我的实现中，$n = 10$。

使用torch的好处是BP算法非常方便，利用torch中的操作构建好网络的前向传播后，torch可以自己推出各个神经元反向传播需要的梯度。

神经网络中还有一些超参数也提前设定了，learning rate = 0.01, max_epoch = 500。学习率的设定也会影响到网络的表现。

可视化

使用Matplotlib来进行可视化。程序运行的最开始，依然是弹出一个窗口，用户可以通过鼠标点击来放置需要拟合的点。放置点结束后，关闭该窗口，程序会弹出一个新的窗口，来动态演示拟合过程。结果如下图：



Log of loss:

```
epoch: 0 loss: 4.429994074627757
epoch: 50 loss: 1.9750015097670257
epoch: 100 loss: 0.9528488150099292
epoch: 150 loss: 0.47010183695056185
epoch: 200 loss: 0.26832549685514095
epoch: 250 loss: 0.17764293453927849
epoch: 300 loss: 0.13121536292783276
epoch: 350 loss: 0.10465543361010532
epoch: 400 loss: 0.08818179870810638
epoch: 450 loss: 0.07733641057410523
```

观看压缩包中的video可以看到动态过程演示。