**ReLU网络拟合函数作业报告**

**2253372 康嘉玮**

1. 题目要求

理论和实践证明，一个两层的ReLU网络可以模拟任何函数[1]。请自行定义一个函数, 并使用基于ReLU的神经网络来拟合此函数。

1. 函数定义与网络构建

2.1 函数定义

我们选择函数y=sin(x)作为模拟对象。

2.2 数据采集

随机选择1000个在[-5, 5)区间上均匀分布的数据作为自变量的值，以它们的正弦值作为函数值。以此我们得到一个训练集。再随机选择100个在[-5, 5)区间上均匀分布的数据作为自变量的值，得到一个大小为训练集1/10的测试集。由于自变量是在[-5, 5]上随机取值的，故能在较大程度上保证训练集与测试集并不相同。

2.3 模型描述

我们选择一个两层的神经网络，隐藏层大小为64。下面代码变量名中的h表示隐藏层。

    self.Wh = np.random.randn(1, 64)

    self.bh = np.random.randn(64)

    self.W = np.random.randn(64, 1)

    self.b = np.random.randn(1)

隐藏层的激活函数选择ReLU。下面的myRelu是自行编写的ReLU函数。

        self.zh = np.dot(x, self.Wh) + self.bh

        self.hidden = myRelu(self.zh) # h = relu(zh)

        self.z = np.dot(self.hidden, self.W) + self.b

损失函数定义为真实值与预测值的差值的平方和的平均数，也就是均方误差。

反向传播算法如下。

        dl\_dz2 = 2 \* (self.z - y) / x.shape[0]

        dl\_dw = np.dot(self.hidden.T, dl\_dz2)

        dl\_db = np.sum(dl\_dz2)

        dl\_dh = np.dot(dl\_dz2, self.W.T)

        dl\_dz1 = dl\_dh \* myRelu\_der(self.zh)

        dl\_dwh = np.dot(x.T, dl\_dz1)

        dl\_dbh = np.sum(dl\_dz1)

        self.W -= learning\_rate \* dl\_dw

        self.b -= learning\_rate \* dl\_db

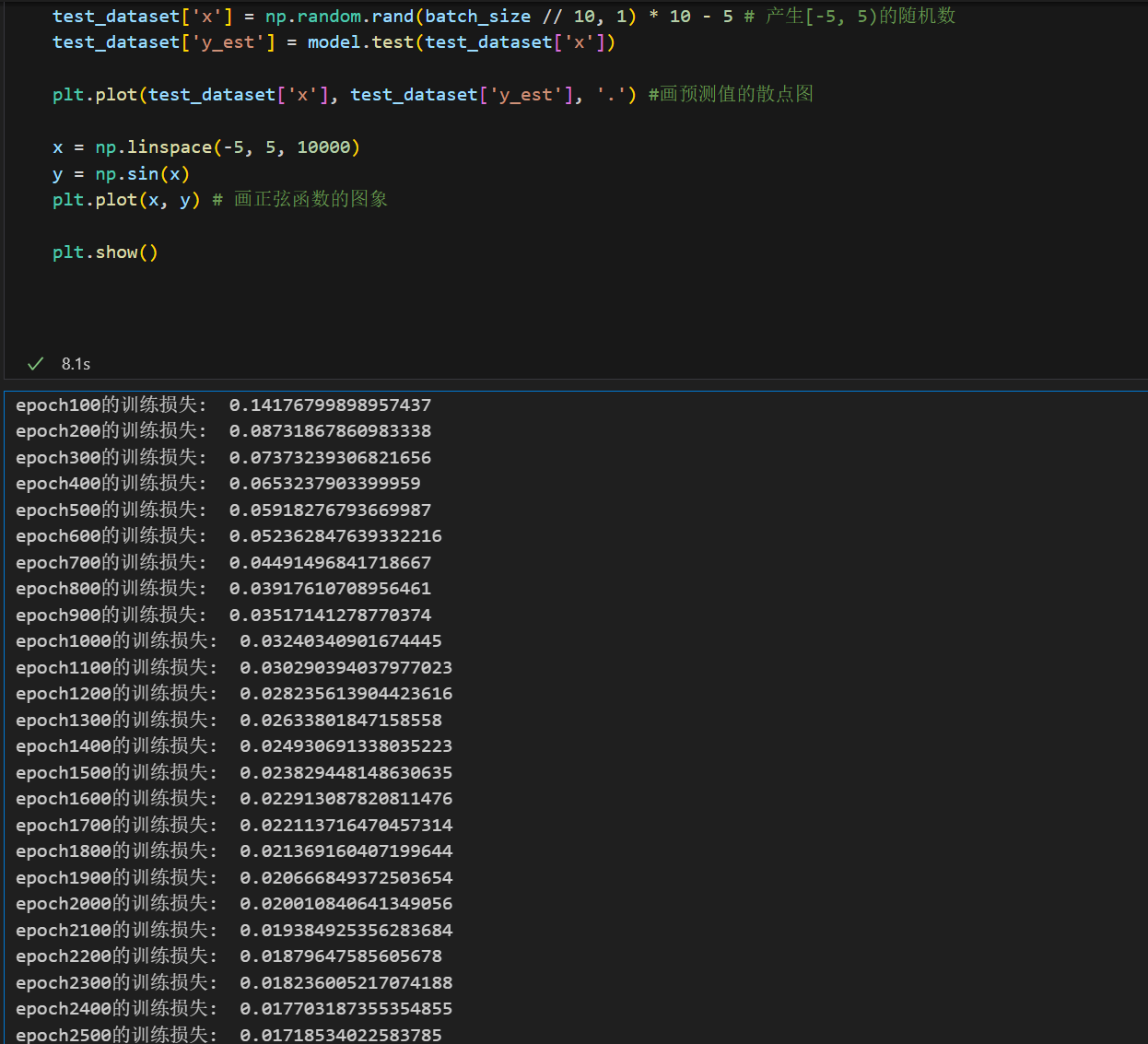
        self.Wh -= learning\_rate \* dl\_dwh

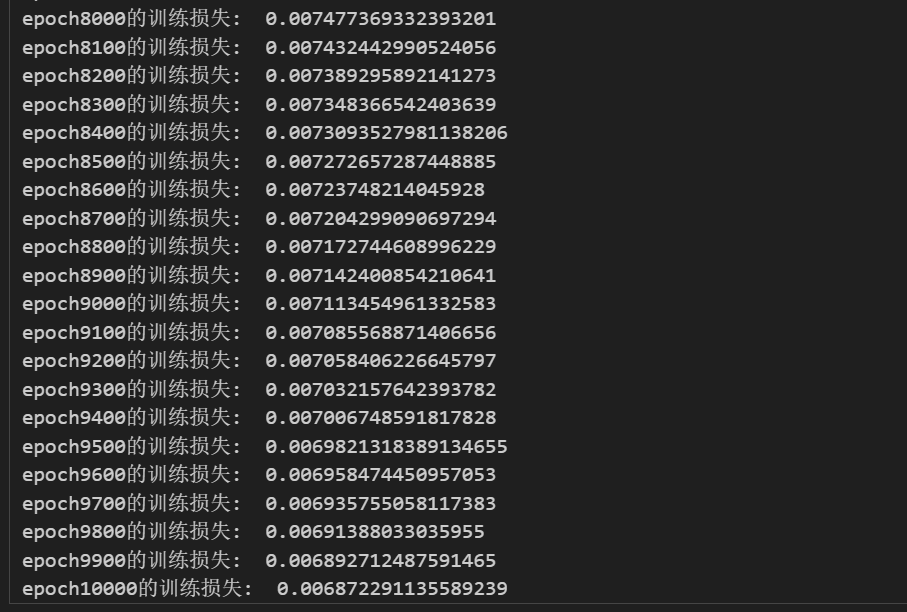
        self.bh -= learning\_rate \* dl\_dbh

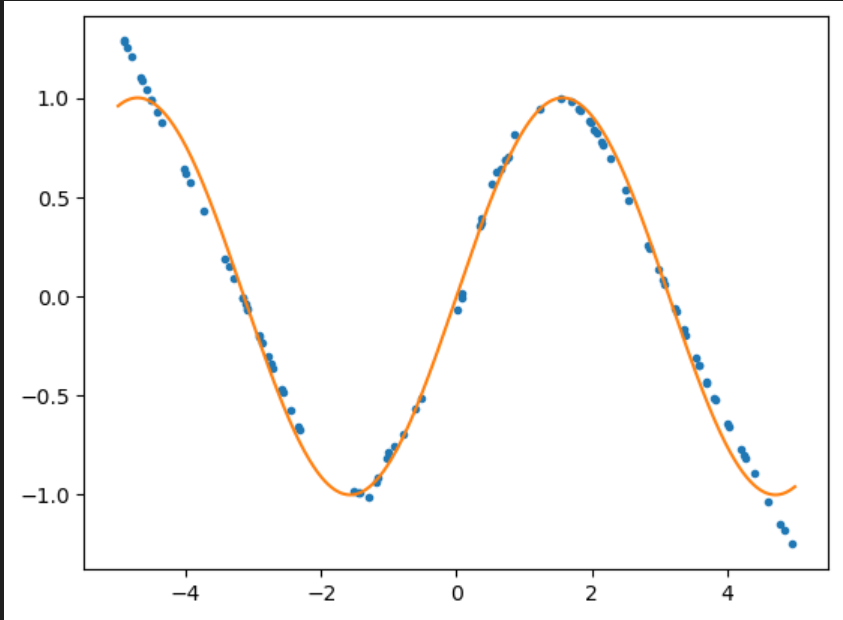
训练完成（10000 epochs）之后，用测试集进行测试，将测试集对应的点显示在一张图上，并与正弦函数的图象进行对比。此处用到了matplotlib库。

1. 拟合效果

运行一次代码，观察拟合效果与最终的图象。下面是部分运行结果图。







由散点图可以得知，拟合的效果较好，大多数散点都大致在正弦曲线上。

1. 收获

通过本次作业，我对ReLU和前馈神经网络的理解加深了许多。希望在日后的学习中，我可以学到深度学习和神经网络方面更多的知识。

参考文献：

1. G. Cybenko. 1989. Approximation by superpositions of a sigmoidal function.
2. K. Hornik, M. Stinchcombe, and H. White. 1989. Multilayer feedforward networks are universal approximators.
3. Moshe Leshno, et al. 1993. Multilayer feedforward networks with a nonpolynomial activation function can approximate any function
4. Vinod Nair and Geoffrey E. Hinton. 2010. Rectified linear units improve restricted boltzmann machines.
5. Xavier Glorot, Antoine Bordes, Yoshua Bengio. 2011. Deep Sparse Rectifier Neural Networks. PMLR 15:315-323.