多项式拟合SINX函数报告

姓名：李欣芮 学号：201822090530

# 实验内容

本次实验主要目的是分析在机器学习、深度学习中，模型的容量与训练集样本大小、目标函数的过拟合和欠拟合以及泛化误差中的方差与偏差之间的影响关系。实验采用多项式拟合sinx函数的实验的过程，通过图像及数据反应以上影响关系。

# 实验方案

实验模型：多项式（构建多项式拟合sinx函数）

实验环境：采用python+tensorflow+numpy+matplotlib环境搭建实验平台。

实验原理：采用该实验环境搭建前馈神经网络，产生的模型则是多项式函数，次数n是模型的容量，训练集根据次数产生相应的样本点，当然在tensorflow中构建一层且一个神经元，该神经元的参数集为，偏值为，激活函数为。随着次数n的增加则神经网络分别代表不同次数的多项式函数，反向传播的损失函数采用MSE( )均方误差函数，参数优化采用梯度下降的学习方式，随着次数的增加，由于激活函数无法避免梯度消失的现象，因此随着次数增加需要对学习率进行减小的调整，已达到更好的训练效果。通过对次数的调整以达到对模型容量的调整同时对训练集大小的调整，以通过matlaplib.pyplot模块进行图像显示实验目的所需分析的过拟合，欠拟合，偏差，方差影响因素。

# 实验步骤

1. 搭建python+tensorflow+numpy+matplotlib的实验环境。
2. 使用numpy模块产生服从高斯分布的32个样本点训练集，并加入的噪声，并绘制出训练集的散点图和y=sinx的函数图像。
3. 搭建前向传播的计算模型（以一次函数为例）。

* 一次函数训练集占位

x1=tf.placeholder(tf.float32,shape=(None,1))

y\_=tf.placeholder(tf.float32,shape=(None,1))

* 前向计算函数

def model(x,w,b):

y=tf.matmul(x,w)+b

return y

* 生成参数

def gw(shape):

w=tf.Variable(tf.random\_normal(shape,mean=0,stddev=1,seed=1))

return w

* 生成偏值

def get\_bias(shape):

b=tf.Variable(tf.random\_normal(shape,mean=0,stddev=1,seed=2))

return b

1. 反向传播

loss1=tf.reduce\_mean(tf.square(y1-y\_))

train\_step1=tf.train.GradientDescentOptimizer(0.001).minimize(loss1)

1. 建立会话初始化变量并进行训练，训练一定(STEPS)轮数。

with tf.Session() as sess:

init\_op=tf.global\_variables\_initializer()

sess.run(init\_op)

STEPS=3000

for i in range(STEPS):

start=i\*BATCH\_SIZE % count

end=i\*BATCH\_SIZE %count + BATCH\_SIZE

sess.run(train\_step1,feed\_dict={x1:X[start:end],y\_:Y\_[start:end]})

if i % 500==0:

print('after',i,' training is:\n')

print('w1:\n',sess.run(w1))

print('b1:\n',sess.run(b1))

line1\_x=np.linspace(0,2\*np.pi,num=50)

print(line1\_x)

line1\_y=sess.run(w1)\*line1\_x+sess.run(b1)

print(line1\_y)

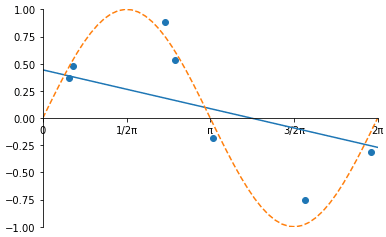
1. 显示最终训练结果

plt.plot(line1\_x,line1\_y[0,:])

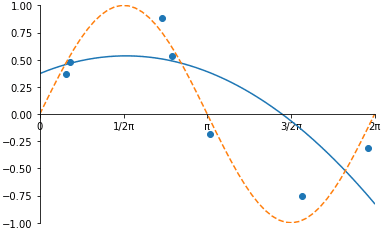
plt.show()

# 实验结果与结果分析

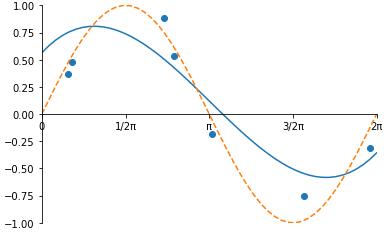
1. 训练集大小为8，一次函数：



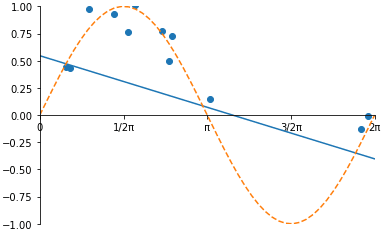
训练集大小为8，二次函数：



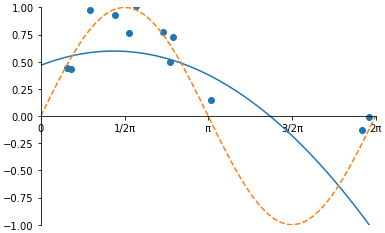
训练集大小为8，三次函数：



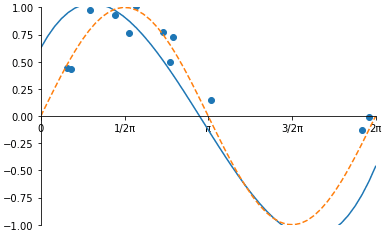
1. 训练集大小为16，一次函数：



训练集大小为16，二次函数：



训练集大小为16，三次函数：



根据均方误差公式以及和上图结果可以得出：训练集不变，均方误差（泛化误差）减小，则随着容量的增加偏差在逐渐减小；容量保持不变，均方误差（泛化误差）减小，则随着训练集数量的增大，方差减小。对于一次函数和二次函数而言容量太小无法拟合sinx函数，欠拟合状态，同样对于三次函数而言8个点的样本集能够很好的拟合，但是对测试集产生的误差较大，因此对于8个点训练集的三次函数而言是处于过拟合状态，因此需要添加更多的样本点以减少泛化误差。