- **第一题：设计海报（15 分）**

  海报详情见附件 1：达特茅斯海报.png

- **第二题：Hello, Machine Learning World. （10 分）**

[1] 读取 MINIST 数据集

```
model = polynomial_model()
# 读取MNIST数据
from tensorflow.examples.tutorials.mnist import input_data
mnist = input_data.read_data_sets("MNIST_data/", one_hot=True)
```

[2] 获得数据集相关参数

```
train_num = mnist.train.num_examples
validation_num = mnist.validation.num_examples
test_num = mnist.test.num_examples
print('MNIST数据集的个数')
print(' >>>train_nums=%d' % train_num,'\n',
      '>>>validation_nums=%d'% validation_num,'\n',
      '>>>test_nums=%d' % test_num,'\n')
```

[3] 获得数据值

```
train_data = mnist.train.images   #所有训练数据
val_data = mnist.validation.images  #(5000,784)
test_data = mnist.test.images       #(10000,784)
print('>>>训练集数据大小：',train_data.shape,'\n',
      '>>>一副图像的大小：',train_data[0].shape)
```

[4] 获得标签值

```
train_labels = mnist.train.labels     #(55000,10)
val_labels = mnist.validation.labels  #(5000,10)
test_labels = mnist.test.labels       #(10000,10)
```

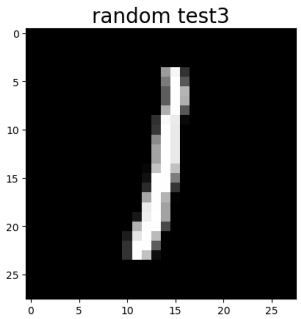
[5] 训练并打印图像

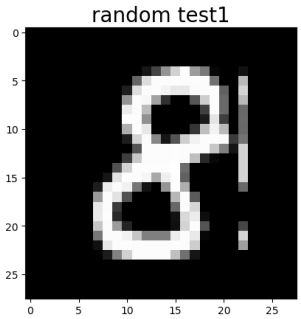
```
plt.figure()
for i in range(10):
    im = train_data[i].reshape(28,28)
    model.train()
    k = random.randrange(9)
    plt.title("random test" + str(k), fontsize=20)
    plt.imshow(im,'gray')
    plt.pause(0.8)
plt.show()
```

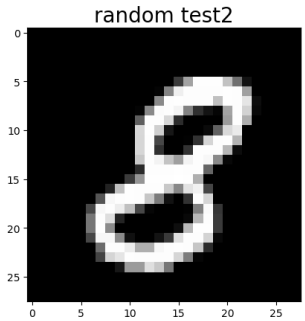
[6] 结果展示

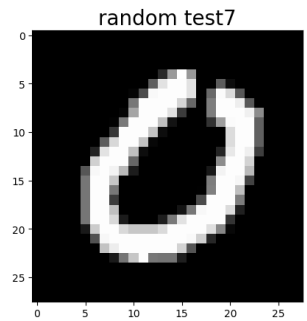
```
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
MNIST数据集的个数
 >>>train_nums=55000
 >>>validation_nums=5000
 >>>test_nums=10000
```

```
>>>训练集数据大小： (55000, 784)
 >>>一副图像的大小： (784,)
Starting Training.............
Sucessfully Finish Training!
Starting Training.............
Sucessfully Finish Training!
Starting Training.............
Sucessfully Finish Training!
Starting Training.............
Sucessfully Finish Training!
```

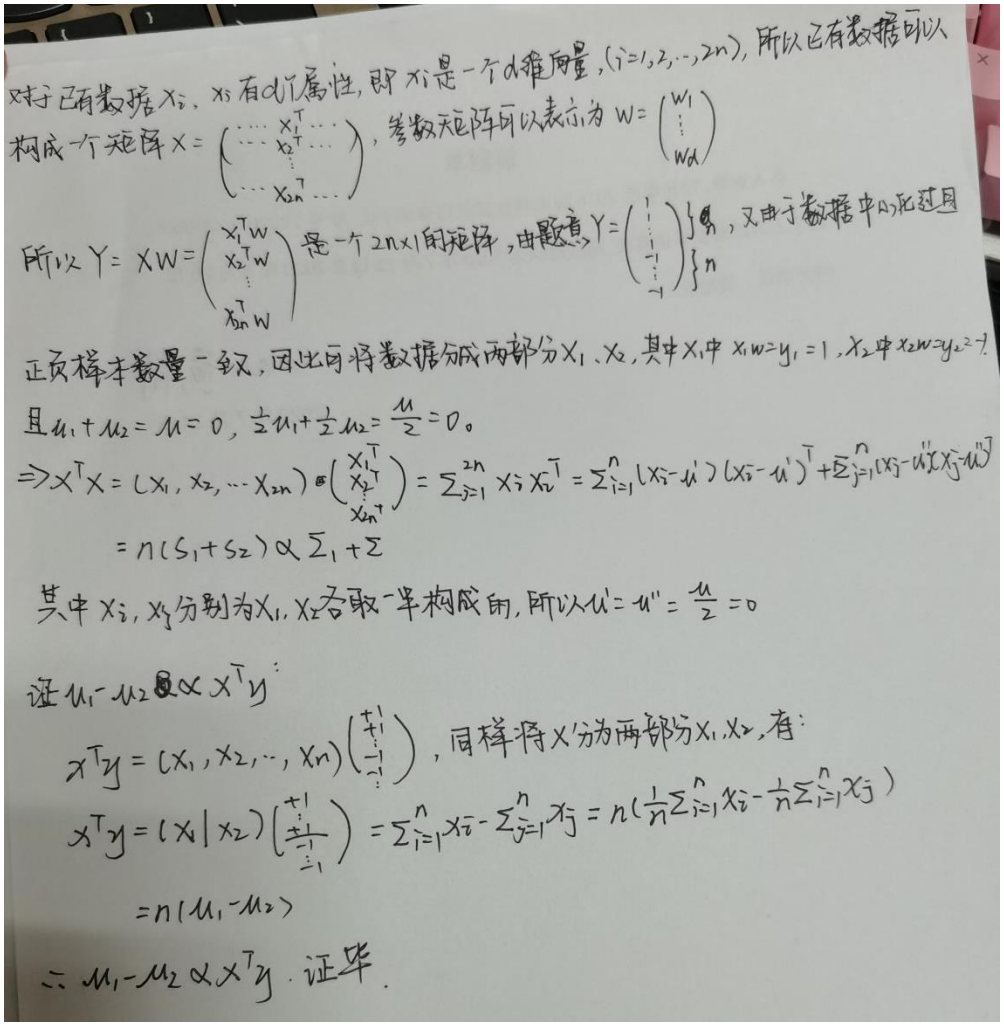








- **第三题：熟练一下矩阵运算（5 分）**



- **第四题： 直线拟合的其他方法（10 分）**

Hough Transform：

Hough 变换是图像处理中从图像中识别几何形状的基本方法之一。Hough 变换的基本原理在于利用点与线的对偶性，将原始图像空间的给定的曲线通过曲线表达形式变为参数空间的一个点。对于直线拟合，在原始图像坐标系下的一个点对应了参数坐标系中的一条直线，同样参数坐标系的一条直线对应了原始坐标系下的一个点，然后，原始坐标系下呈现直线的所有点，它们的斜率和截距是相同的，所以它们在参数坐标系下对应于同一个点。这样在将原始坐标系下的各个点投影到参数坐标系下之后，看参数坐标系下有没有聚集点，这样的聚集点就对应了原始坐标系下的直线。

RANSAC

随机抽样一致（RANSAC）是一种通过使用观测到的数据点来估计数学模型参数的迭代方法。这是一种非确定性算法，因为它是在一定概率下得到一个合理的结果，当迭代次数增加，概率也会增加。在两个方向上拟合直线，其中包括 inlier（可以被拟合直接），outlier（不可以

拟合直线)，最小二乘法这里便不适用，因为它会为拟合所有点，而 RANSAC 却只仅适用 inlier 来计算拟合。这由一些数据的随机样本来拟合线性模型以及找到其中有最好拟合的数据子集。

优缺点

优点：Hough 变换能够查找任意的曲线，只要你给定它的方程。Hough 变换在检验已知形状的目标方面具有受曲线间断影响小和不受图形旋转的影响的优点，即使目标有稍许缺损或污染也能被正确识别；RANSAC 算法的有点事鲁棒性好，即使有不小程度的 outliers 存在也能够估计模型参数。缺点便是其计算参数没有时间上限（除了“疲劳”）当迭代次数被限制时，得到的可能便不是最优的，甚至可能很差；通过目标函数最优化求解参数，能够较容易地求解目标函数的极值，使得训练样本的损失最小化。

缺点：Hough 变换算法的特点导致其时间复杂度和空间复杂度都很高，并且在检测过程中只能确定直线方向，丢失了线段的长度信息；一般的 RANSAC 的权衡是：随着计算次数增加，模型一般会更好。但 RANSAC 并不能一直寻找到最优集（但当 inliers 小于 50%的时候往往表现很差），其次 RANSAC 必须要设置一个阈值，最后，RANSAC 一次只能估计出一个模型，如果两个（或更多）的模型例子存在，RANSAC 并不能寻找到；目标函数最优化求解参数在某些情况下可能不能正确的找到各种情况下的极值，因为某些目标函数无法求导，而且因为需要进行大量的数值运算，某些情况下算法的速度并不快。

- **第五题： 根据实验结果写论文。（60 分）**

（见下页）

厦门大学

XIAMEN UNIVERSITY

基于多项式的曲线拟合方法与分析

黄静

厦门大学计算机系

golden_jing@qq.com

摘要

在机器学习算法中，基于针对数据的非线性函数的线性模型是非常常见的，这种方法即可以像线性模型一样高效的运算，同时使得模型可以适用于更为广泛的数据上，多项式拟合就是这类算法中最为简单的一个。本文通过使用多项式对有噪音的数据进行曲线拟合，并对阶数、噪声对曲线拟合的影响进行分析。该案例涉及到了机器学习的几大要素，包括数据、模型和算法，可有效加深对 ML = Loss + Algorithm + Model + BigData + Application 的理解。同时通过对模型参数的选择，本次实验中即对多项式阶数的选择，以及对过拟合的分析和通过正则化解决过拟合问题，使得该机器学习模型更完善。

Abstract

In machine learning algorithms, linear models based on non-linear functions for data are very common. This method can operate as efficiently as linear models, and at the same time makes the model applicable to a wider range of data, polynomial fitting It is the simplest one of this type of algorithm. This paper uses polynomials to fit the noisy data, and analyzes the influence of the order and noise on the curve fitting. This case involves several major elements of machine learning, including data, models and algorithms, which can effectively deepen the understanding of ML = Loss + Algorithm + Model + BigData + Application. At the same time, through the selection of model parameters, the selection of polynomial order in this experiment, the analysis of overfitting and the solution of overfitting through regularization, the machine learning model is more perfect.

引言

曲线拟合(Curve Fitting)的数学定义是指用连续曲线近似地刻画或比拟平面上一组离散点所表示的坐标之间的函数关系，是一种用解析表达式逼近离散数据的方法。曲线拟合通俗的说法就是“拉曲线”，也就是将现有数据透过数学方法来代入一条数学方程式的表示方法。科学和工程遇到的很多问题，往往只能通过诸如采样、实验等方法获得若干离散的数据，根据这些数据，如果能够找到一个连续的函数（也就是曲线）或者更加密集的离散方程，使得实验数据与方程的曲线能够在最大程度上近似吻合，就可以根据曲线方程对数据进行数学计算，对实验结果进行理论分析，甚至对某些不具备测量条件的位置的结果进行估算。本次实验中讨论的多项式曲线拟合（Polynomial Curve Fitting）是曲线拟合中最常见的一种形式，通常用于多项式回归的应用中，是线性回归模型中的一种。

多项式模型与算法求解

模型

首先假设我们的训练集由x的N次观测得到，为$x_1,x_2,x_3,\dots,x_n$均匀分布于区间[0,1]。对应的观测集为$y_1,y_2,y_3,\dots,y_n$，实际的目标函数为f(x)。其中训练集是已知的，我们的目的是通过训练集和观测集拟合出预测函数，让它尽可能的接近目标函数，所以生成观测集的方式是通过训练集加上随机噪声输入到目标函数得到。因为在现实问题中获取到的训练数据也是夹杂了各种噪声，所以这样的拟合对于实验更有意义。

随后我们需要使用多项式来拟合我们的数据，多项式定义为：

$$\mathbf{y(x,W)} = \boldsymbol{w_0} + \boldsymbol{w_1 x} + \boldsymbol{w_2 x^2} + \cdots + \boldsymbol{w_M x^M} = \sum_{\boldsymbol{j=0}}^{\boldsymbol{M}} \boldsymbol{w_j x^j}$$

其中 $\mathbf{M}$ 是多项式的阶数，$\boldsymbol{w_0,\dots,w_M}$是多项式的系数，记做$\mathbf{W}$，可以看到虽然多项式函数$\mathbf{y(x,W)}$是关于x非线性函数，但是却是关于多项式系数$\mathbf{W}$的线性函数。有了多项式以后我们还需要一个误差函数来对我们拟合出的多项式进行评估，这里我们使用均方误差，公式如下：

$$\mathbf{E(W)} = \frac{\mathbf{1}}{\mathbf{2}} \sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} \{\boldsymbol{y(x_n,W)} - \boldsymbol{t_n}\}^2$$

当我们一个模型的复杂度给定了之后，数据规模增加能够有效的减轻模型的过拟合问题，所以这也是一个防止模型过拟合的方式。除了通过增加数据量的方式来减轻过拟合的影响，还可以通过正则化的方式来实现，这种技术就是在我们定义为误差函数增加一个惩罚项，是的多项式系数被有效的控制。均方误差函数修改后的形式为：

$$\boldsymbol{\tilde{E}(x)} = \frac{\mathbf{1}}{\mathbf{2}} \sum_{\boldsymbol{n=1}}^{\boldsymbol{N}} \{\boldsymbol{y(x_n,W)} - \boldsymbol{t_n}\}^2 + \frac{\boldsymbol{\lambda}}{\mathbf{2}} \boldsymbol{||W||^2}$$

其中$\boldsymbol{||W||^2} = \boldsymbol{W^T W} = \boldsymbol{w_0}^2 + \boldsymbol{w_1}^2 + \cdots + \boldsymbol{w_M}^2$，这种二次正则项的应用也叫作山脊回归（Ridge Regression）。

算法

根据模型中的解释可以看出，对于不包含惩罚项的误差函数$\mathbf{E(W)}$可以表示为：

$$\mathbf{E(W)} = \frac{\mathbf{1}}{\mathbf{2}}\sum_{n=1}^{N}\{\boldsymbol{y}(\boldsymbol{x_n}, \boldsymbol{W}) - \boldsymbol{t_n}\}^{\mathbf{2}} = \frac{\mathbf{1}}{\mathbf{2}}(\boldsymbol{XW} - \boldsymbol{T})^{\boldsymbol{T}}(\boldsymbol{XW} - \boldsymbol{T})$$

对其求导得:

$$\frac{\boldsymbol{\vartheta E(W)}}{\boldsymbol{\vartheta W}} = \boldsymbol{X^T XW} - \boldsymbol{X^T T}$$

令其导数等于 0，得:

$$\mathbf{W} = (\boldsymbol{X^T X})^{-\mathbf{1}}\boldsymbol{X^T T}$$

这样就可以通过矩阵运算直接得出$\mathbf{W}$。对于包含惩罚项的误差函数$\widetilde{\boldsymbol{E}}(\mathbf{W})$，对其求导可得:

$$\frac{\boldsymbol{\vartheta}\widetilde{\boldsymbol{E}}(\boldsymbol{W})}{\boldsymbol{\vartheta W}} = \frac{\mathbf{1}}{\boldsymbol{m}+\mathbf{1}}((\boldsymbol{X^T X} + \boldsymbol{\lambda E_{m+1}})\boldsymbol{W} - \boldsymbol{X^T X})$$

求出$\mathbf{W}$:

$$\mathbf{W} = (\boldsymbol{X^T X} + \boldsymbol{\lambda E_{m+1}})^{-\mathbf{1}}\boldsymbol{X^T T}$$

这样同样可以直接由矩阵运算求得$\mathbf{W}$。

实验结果与分析

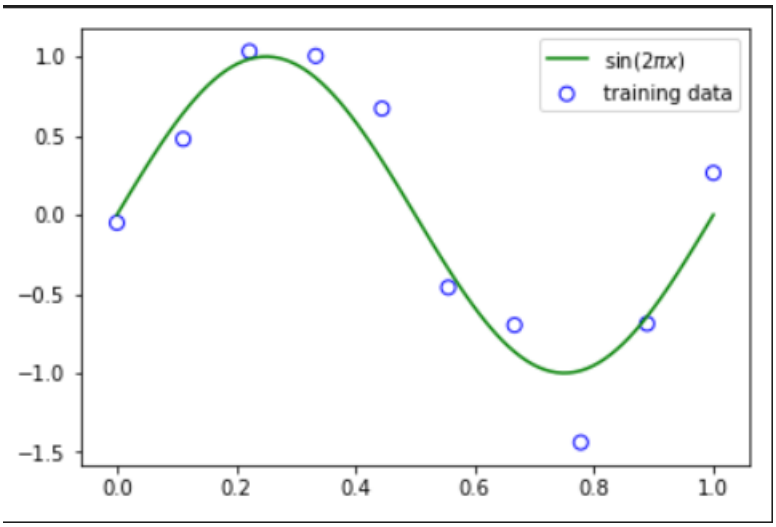
数据集的产生

本次实验的数据集中于均匀分布在区间[0,1]之间，且每个数据随机加上服从高斯分布的噪声，实际拟合的函数为$\sin(2\pi x)$，即每个数据的实际值为在该点的$\sin(2\pi x)$的值加上随机噪声值。

实验 1: 生成初始数据的分析

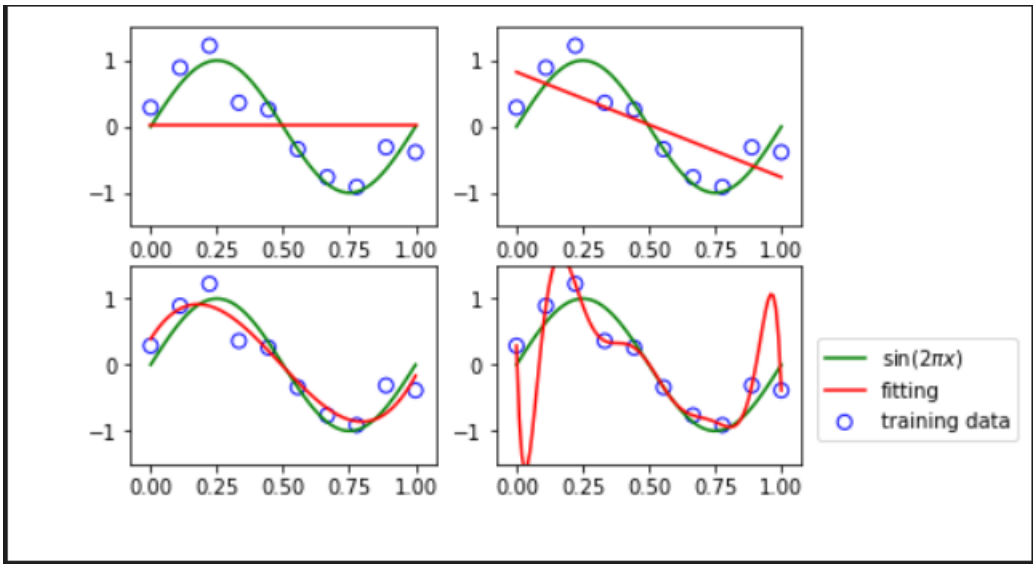
```
def create_toy_data(func, sample_size, std):
    x = np.linspace(0, 1, sample_size)
    t = func(x) + np.random.normal(scale=std, size=x.shape)
    return x, t

def func(x):
    return np.sin(2 * np.pi * x)

x_train, y_train = create_toy_data(func, 10, 0.25)
x_test = np.linspace(0, 1, 100)
y_test = func(x_test)

plt.scatter(x_train, y_train, facecolor="none", edgecolor="b", s=50, label="training data")
plt.plot(x_test, y_test, c="g", label="$\sin(2\pi x)$")
plt.legend()
plt.show()
```

其中调用了 numpy 库中的 numpy.random.normal 函数来生成符合高斯分布、方差为 0.25 的随机噪声，数据量为 10，均匀分布在[0,1]之间，即数据间隔为 0.1，同时画出[0,1]上的$\sin(2\pi x)$函数，结果为:



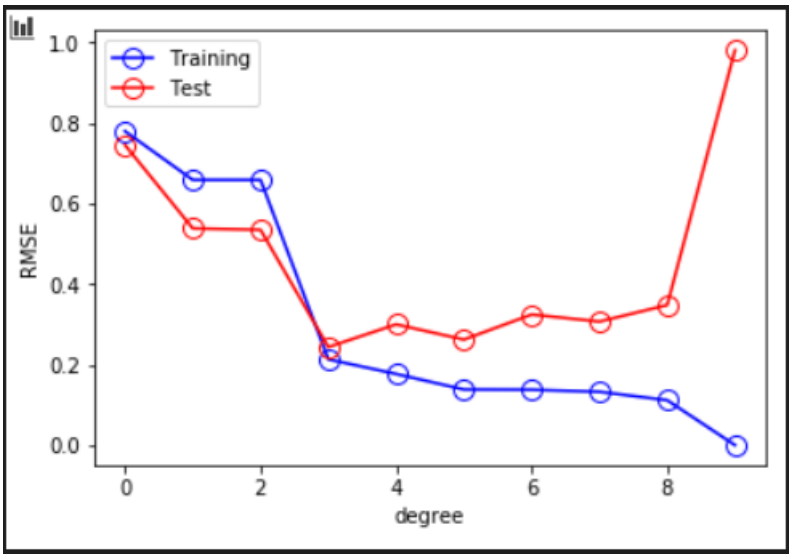
实验 2: 对拟合多项式阶数作为参数的分析

通过调整多项式函数的最高阶数，对不同阶数拟合的情况进行讨论。本实验中测试多项式阶数 M = 0，1，3，9 情况下的拟合情况，并给出四个拟合多项式结果。结果如下所示:



从图中可以看出，红色曲线就是我们的预测函数，可以看到在 M=0 和 M=1 的时候，多项式对于数据的拟合效果极差，不能很好的代表我们的目标函数，这种情况下为欠拟合。当 M=3 时多项式已经非常接近我们的目标函数$\sin(2\pi x)$了。当 M=9 时，多项式函数精确的通过了每一个数据点，这时$E(W) = 0$,拟合曲线呈现震荡形式并且对噪声数据敏感，但是这种情况过拟合。

为了测试在不同阶数下模型的性能，我们对阶数不同的模型对于新的测试数据的预测性能进行测试，阶数 M = 0,1,2,⋯,9 共十种情况并用相同的测试数据对误差进行分析，结果如下：

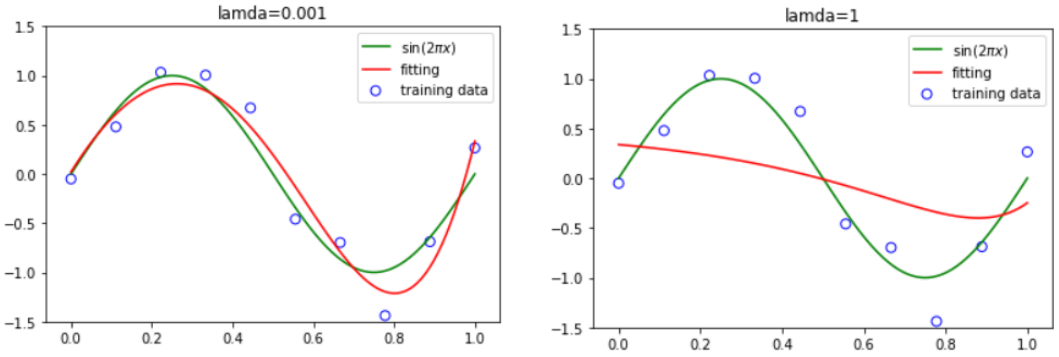


测试集的误差衡量了对于新观察到的数据 x，我们预测 t 的值的效果的好坏。根据上图结果可以看出，我们看到小的 M 值会造成较大的测试集误差，这可以归因于对应的多项式函数相当不灵活，不能够反映出$\sin(2\pi x)$的震荡。当 M 的取值为 3 ≤ M ≤ 8 时，测试误差

较小，对于生成函数$\sin(2\pi x)$也能给出合理的模拟。

实验 3：正则化参数的分析

除了通过增加数据量的方式来减轻过拟合的影响，还可以通过正则化的方式来实现，这种技术就是在我们定义为误差函数增加一个惩罚项，是的多项式系数被有效的控制。正则化参数中最重要的为惩罚项系数λ，我们在 M=9 的情况下，原本会出现严重的过拟合现象，现在引入λ=1 和λ=0.001 两种情况，下面展示这两种情况下过拟合被压制的情况：



从图中可以看出，当λ=0.001 时，过拟合现象被压制，此时我们可以得到关于实际拟合函数$\sin(2\pi x)$的一个更好的模拟；但是当λ过大的时候，我们又会得到一个不好的结果，即一个过度抑制模型，所以根据模型复杂度来选择合适的λ对于模型最终的拟合结果有着重要的影响。

实验结果总结

在多项式拟合曲线的过程中，如果当我们模型复杂度被限制了，我们可以通过增加训练数据的方式防止模型的过拟合以及减轻被噪声数据的影响。同样当我们数据有限，想通过分析数据来选择模型的复杂度的话也可以通过正则化的方式来抑制模型的过拟合。当然也可以同时使用两种方式，合理的调整才能达到最佳拟合。