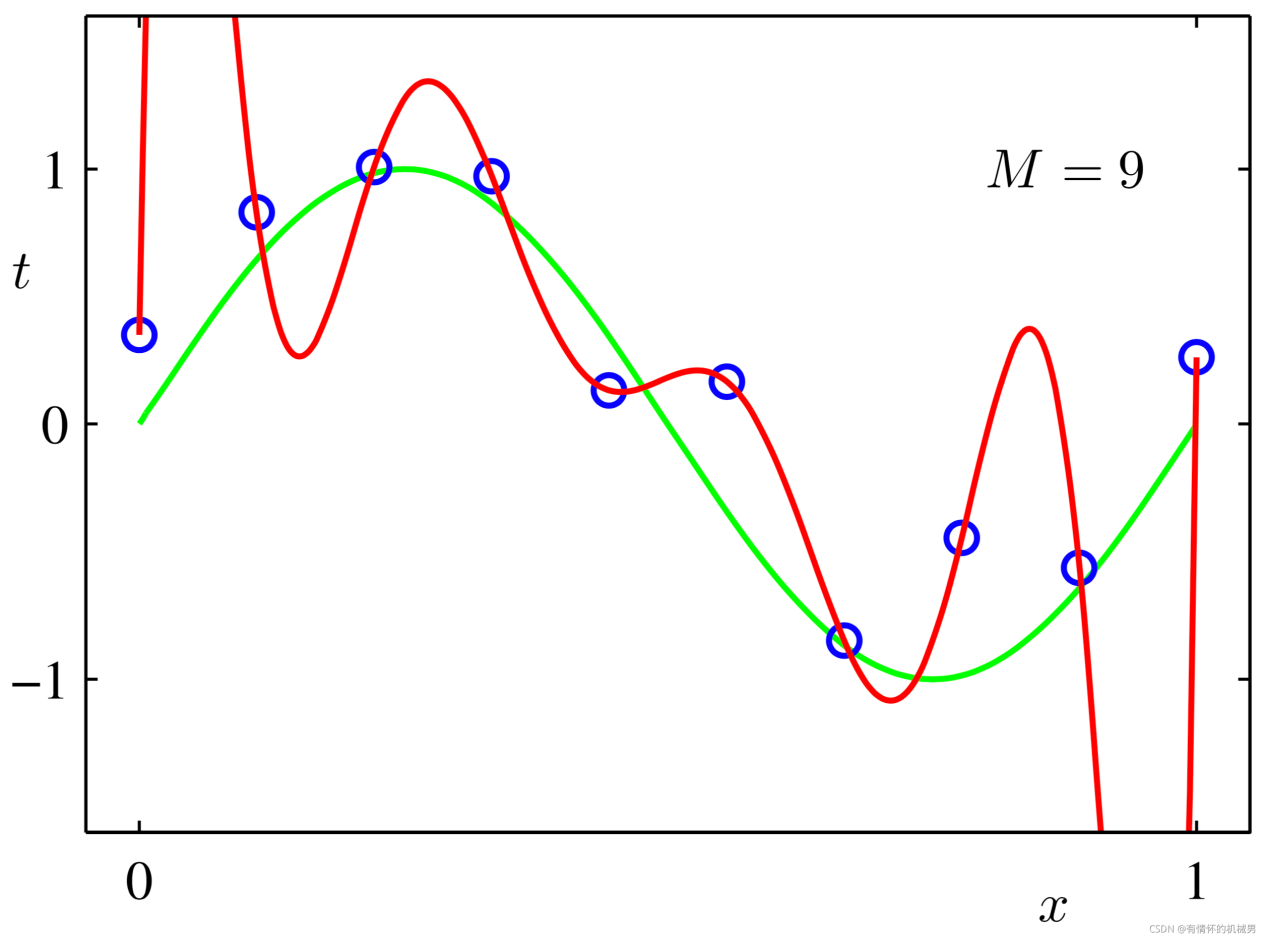
**过拟合**

过拟合其实就是为了得到一致假设而使得假设过于地严格。使得其在训练集上的表现非常地完美，但是在训练集以外的数据集却表现不好。



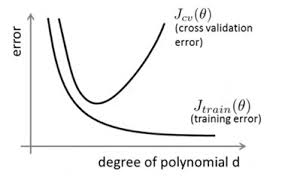
如上图所示，红线就是过拟合了，虽然它在训练集上将所有的点都放在了线上，但是如果再来一个点就会不起作用，这就是过拟合，而绿线的话也比较好地拟合了点集，但是它的泛化能力相较于红线来说是更好的。

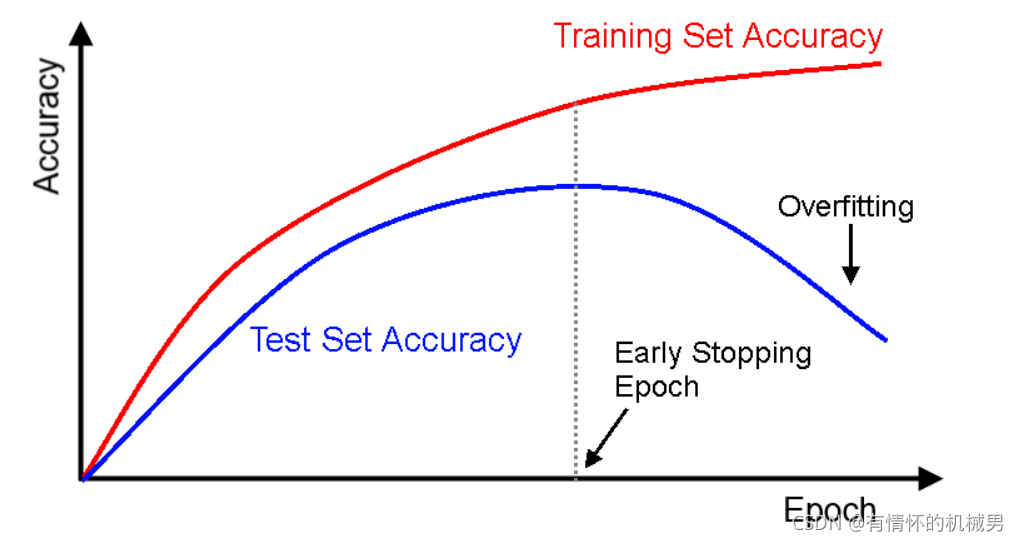
## **过拟合的表现（判定方法）**

1、训练集的正确率不增反减

2、验证集的正确率不再发生变化

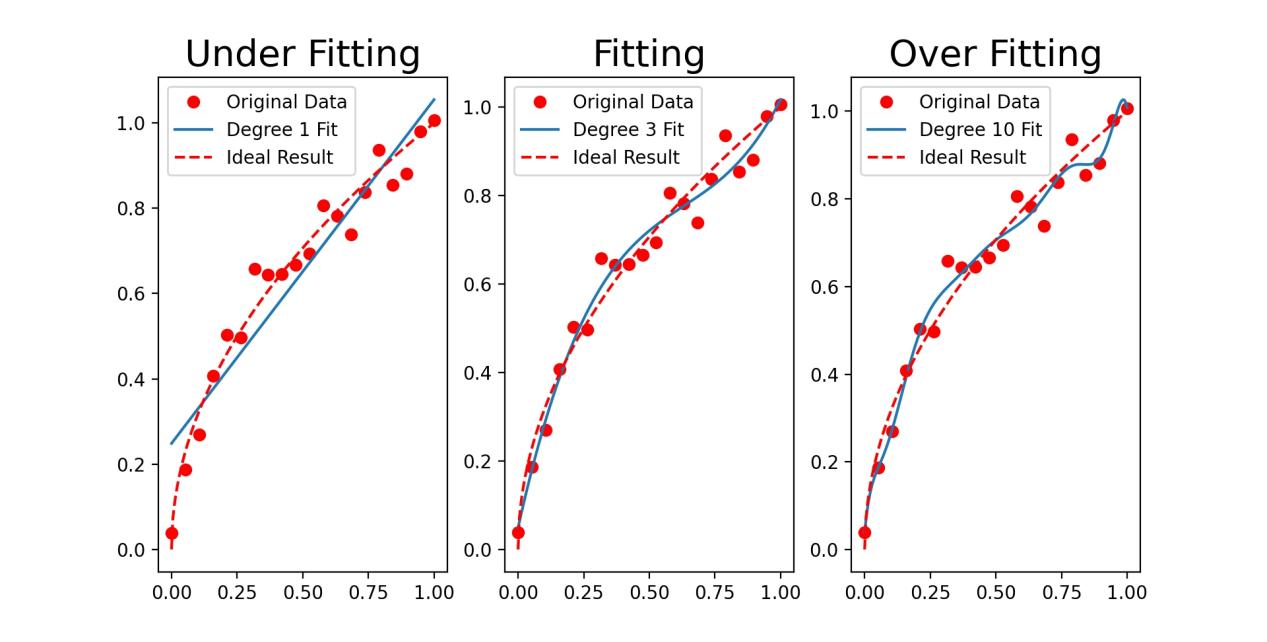
3、训练集的error一直下降，但是验证集的error不减反增





代码





## **产生过拟合的原因**

**1、样本方面**

1）用于训练的样本量**过于少**，使得训练出来的**模型不够全面**以致于使用模型时错误率高。比如对于猫这类动物，如果训练数据集中只有一个正拍且坐立的猫，那么当过拟合时，模型往往有可能只能识别出这类姿态的猫，像跳跃的猫、局部捕捉的猫、反转的猫等等可能都识别不出来了

2）训练的样本量**噪声大**（质量不好），导致一些错误的特征错认为是学习的对象，使得训练模型不够健壮

**2、模型方面**

1）参数过多，模型过于复杂

2）选择的模型本身就不适用于当前的学习任务

3）网络层数过多，导致后面学习得到的特征不够具有代表性

## **避免过拟合的方法**

**1、样本方面**

**1）增加样本量**

深度学习中样本量一般需要在万级别才能训练出较好的模型，且样本尽可能地多样化，使得样本更加地全面。一般通过图像变换可以进行数据增强。

**2）样本筛选**（特征降维、特征选择）

特征降维：PCA等，根据现有的特征创造出新的特征

特征选择：选择具有代表性的特征参与训练、

**3）归一化数据样本**

改变数据的分布，使得更集中在激活函数的敏感区

1. **模型方法**

**正则化——使模型简单化、参数稀疏化**

概念——核心思想

软正则化其实就是在原来的**损失函数**后面增加了一项**加数**，这项加数称之为正则项，这个正则项通常**由一个系数和范数**的乘积的累加和构成。其主要是通过正则项来限制**权重ω参数**的值的变化，使其尽可能的**小**或者尽可能地**趋于0**，以达到**稀疏参数**的效果，进而使得模型**复杂度下降**，避免过拟合。

/\*问题1：什么是稀疏参数？什么是数据的稀疏性？

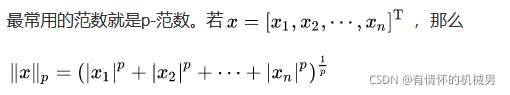
稀疏参数：使得模型中参数的零分量尽可能地多；

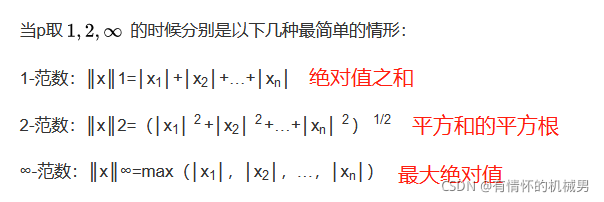
数据的稀疏性：在特征选择中的概念，指的是在众多的特征中，有的特征对于模型的优化是无关的，这就是数据存在稀疏性，可以通过特征选择来选择有价值的特征。

但是在机器学习的参数中，由于参数众多，模型复杂度过于高的话，易出现过拟合现象，因此我们需要通过增加参数的稀疏性来降低复杂度，进而避免过拟合。\*/

/\*问题2：什么是范数？常用的范数函数是什么？

范数：在泛函分析中，它定义在赋范线性空间中，并满足一定的条件，即①非负性；②齐次性；③三角不等式。它常常被用来度量某个向量空间（或矩阵）中的每个向量的长度或大小



\*/

/\*问题3：实现参数的稀疏有什么好处吗？

一个好处是可以简化模型，避免过拟合。因为一个模型中真正重要的参数可能并不多，如果考虑所有的参数起作用，那么可以对训练数据可以预测的很好，但是对测试数据表现性能极差。另一个好处是参数变少可以使整个模型获得更好的可解释性。\*/

/\*问题4：参数值越小代表模型越简单吗？

是的。为什么参数越小，说明模型越简单呢，这是因为越复杂的模型，越是会尝试对所有的样本进行拟合，甚至包括一些异常样本点，这就容易造成在较小的区间里预测值产生较大的波动，这种较大的波动也反映了在这个区间里的导数很大，而只有较大的参数值才能产生较大的导数。因此复杂的模型，其参数值会比较大。反过来就是如果参数值小的话，那么异常点在这个区间里的导数就会比较小，造成预测值的波动也就小，这样就会利于模型，避免过拟合，更具有泛化能力去预测新样本。\*/

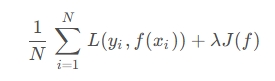
正则化（regularization）的基本形式是在模型的损失函数中添加一个**正则化项**（regularizer）或**惩罚项**（penalty term）

微信截图_20250626131140

L 代表**损失函数**，用于度量模型预测与实际标签之间的**差距**；

J 是**正则化项**，用于约束模型的**复杂度**；

λ 是**正则化系数**，用于调控损失函数和正则化项之间的**权衡**。



正则化项通常是模型参数的 **L1 范数**或 **L2 范数**。具体来说，添加模型参数的 L1 范数的正则化被称为 L1 正则化，而添加模型参数的 L2 范数的正则化被称为 L2 正则化。在某些情况下，可能会结合 L1 和 L2 范数，形成所谓的“弹性网络”（Elastic Net）正则化项，这是一种折中策略。

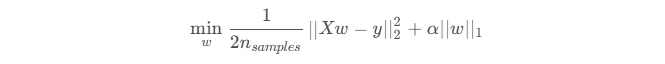
**L1 范数**：也称为曼哈顿距离，它是模型参数绝对值的和。L1范数有一个有趣的特性，即它倾向**于产生稀疏的权重矩阵**，即其**许多元素都为零**。在机器学习中，这可以被用作一种自动特征选择机制。

**L2 范数**：也称为欧几里得距离，它是模型参数平方和的平方根。相比之下，L2范数倾向于**分散权重值**，即它更倾向于平均地分配权重值，而不是集中在少数几个特征上。

**L1正则化（LASSO）——效果使ω分量往0靠拢**

Lasso（Least Absolute Shrinkage and Selection Operator）也是一种改进的线性回归方法，通过在损失函数中加入L1正则化项，来实现特征选择以及防止过拟合。L1正则化项即模型**权重参数的绝对值和**，其目标在于**生成稀疏的权重**，即许多权重为0，以此达到特征选择的目的。

Lasso回归的优化目标可以用以下公式表示：



在此公式中，目标是找到一组**权重参数ω**，使得模型预测值和真实值之间的**误差与正则化项的总和最小**。公式的第一部分微信截图_20250626133248是模型预测值与真实值之间的**均方误差**（还除以了样本数量的一半，保证损失不会因样本数量的变化而显著改变），这是主要的**优化目标**。公式的第二部分 微信截图_20250626133359 是**L1正则化项**，**α是一个超参数**，用于控制正则化的强度。

在Lasso模型中，选择合适的α值是关键步骤。**α值越大，正则化效果越强，产生的稀疏权重也越多，即选择的特征越少**。反之，α值越小，正则化效果越弱，模型可能选择更多的特征，进而更可能产生过拟合。

**L2正则化（岭回归）——效果使参数减小**

岭回归是线性回归的一种改进，它通过在损失函数中添加一个L2正则化项来防止过拟合。L2正则化项是**模型权重参数的平方和**，它的目标是**限制模型权重的大小**，防止模型过于复杂。

岭回归的优化目标可以通过下面的公式来描述：

微信截图_20250626133911

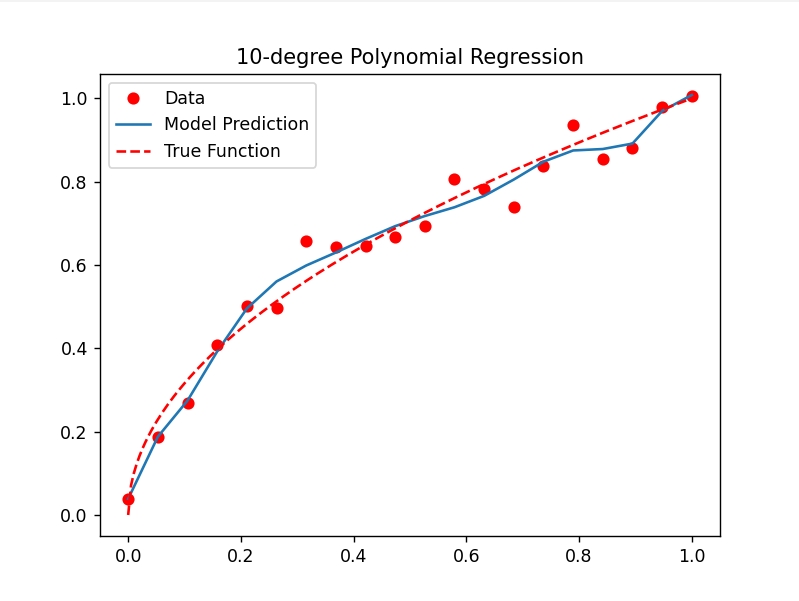
这个公式的目标就是希望找到一个权重参数ω，使得模型的**预测误差和正则化项的和最小**。公式的第一部分 微信截图_20250626133248 是模型的预测值与真实值之间的**均方误差**，这是主要优化目标。公式的第二部分 微信截图_20250626134031是**L2正则化项**，**α是一个超参数**，用于控制正则化的强度。

选择适当的 α值是岭回归模型中的一个关键步骤。**α值越大，正则化的效果越强，模型的复杂度就越低，防止过拟合的能力就越强**。相反，α值越小，正则化的效果就越弱，模型就越可能过拟合。

Step 1：在原始数据中衍生出几个特征，分别是微信截图_20250626134555然后带入线性回归方程进行建模

代码





**红色的点**代表了生成的**实际数据点**，**蓝色的线**代表了**使用10阶多项式特征进行线性回归后的预测结果**，而**红色的虚线代表了理想的预测结果**。数据是基于微信截图_20250626135208这个函数生成的，这是一个单调递增的平滑函数。而使用的模型是一个10阶的多项式模型，这是一个非常复杂的模型，可以拟合出复杂的非线性关系。

从图中可以看出，**蓝色的线（模型预测结果）在大部分地方都非常接近红色的点（实际的数据点）**，甚至在一些地方，蓝色的线几乎完全覆盖了红色的点，这**说明模型在训练数据上的表现非常好**，几乎可以完美地拟合所有的训练数据。

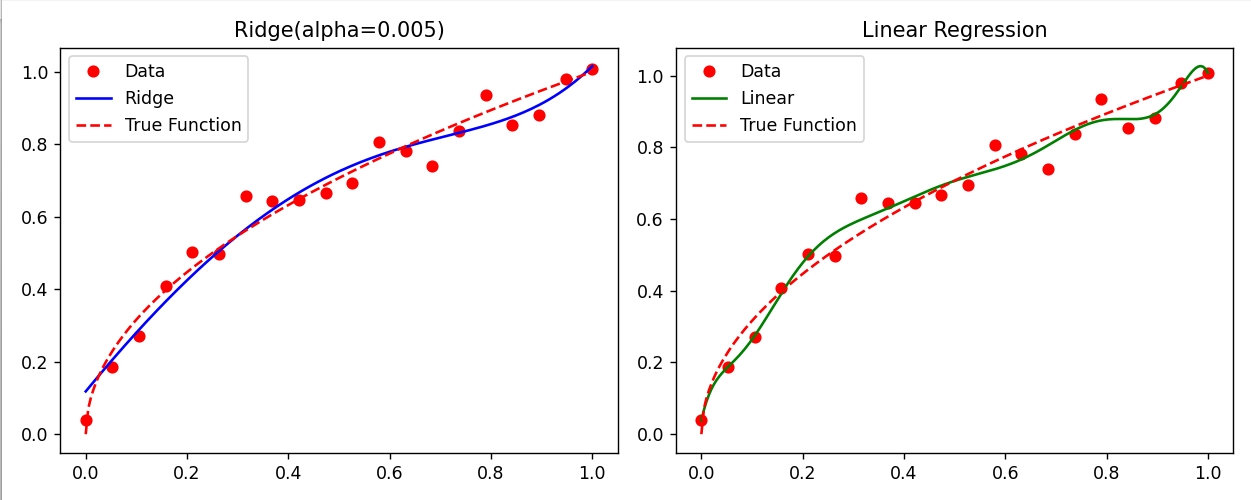
然而，当观察**蓝色线与红色虚线（理想的预测结果）的关系**时，会发现，**蓝色的线在一些地方与红色虚线有着明显的偏差**。特别是在x值较大或较小的地方，蓝色线的波动非常大，明显偏离了理想的预测结果。

这就说明，**模型在训练数据上的表现虽然非常好，但是在理想的预测结果上的表现却不尽人意，这是典型的过拟合现象**。也就是说，模型过度地**学习了训练数据中的噪声，而忽视了数据的真实分布**，这导致模型在新的、未见过的数据上的表现可能会非常差。

Step 2: 加入正则化项Ridge

针对上述的过拟合数据，尝试在线性回归的损失函数中引入正则化，来缓解10阶特征衍生后的过拟合问题。先试一下岭回归。





左边的图像是应用了岭回归（Ridge）之后的结果，右边的图像是应用了普通的线性回归（Linear Regression）的结果。

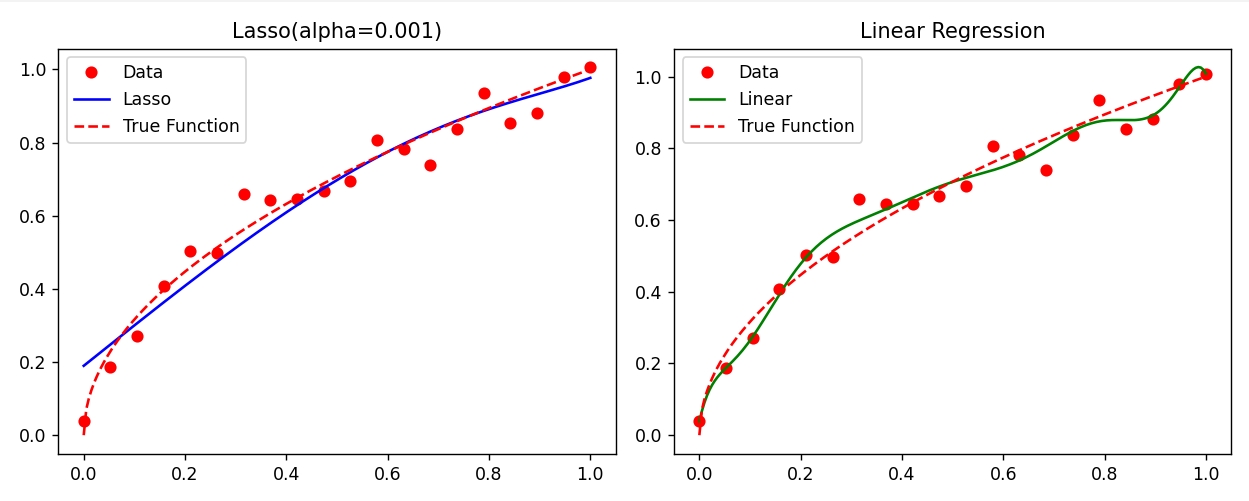
岭回归是一种改良的最小二乘估计法，通过对系数的大小施加惩罚来解决普通最小二乘法在自变量存在多重共线性时可能出现的过拟合问题。也就是说，通过引入一个α参数（也就是代码中的0.005），岭回归在尽可能保持系数总体规模不变的前提下，对系数进行调整，使得预测结果更为稳定。

从图中可以看出，使用岭回归之后，模型的**预测结果（左图中的蓝色线）相较于普通的线性回归（右图中的绿色线），在x值较大或较小的地方，偏离实际数据点（红色点）的程度有所降低**，也更接近理想的预测结果（红色虚线）。这说明引入岭回归后，模型的过拟合现象得到了一定程度的缓解。

值得注意的是，岭回归的**α参数的选择对模型的效果有很大影响**，α过大可能会导致模型欠拟合，α过小又可能收效不明显。因此在实际使用中，通常需要通过交叉验证等方式来选择一个合适的α值。

Step 3 ：加入正则化项Lasso





从图像中看到：**Lasso的惩罚力度更强**，并且迅速将一些参数清零，而这些被清零的参数，则代表对应的参数在实际建模过程中并不重要，从而达到特种重要性筛选的目的。在实际的建模过程中，**L2正则化往往应用于缓解过拟合趋势，而L1正则化往往被用于特征筛选的场景中。**

**硬正则化(Hardware Regularization, HR)**是一种在深度学习模型训练过程中，通过引入额外的**约束条件来限制模型权重**的方法。这种方法的目的是提高模型的泛化能力，减少过拟合问题。硬正则化不同于软正则化(Software Regularization, SR)，软正则化通过调整**损失函数**中的正则项来实现模型的简化，而硬正则化通过直接**限制模型的结构**或**参数**来实现模型的简化。

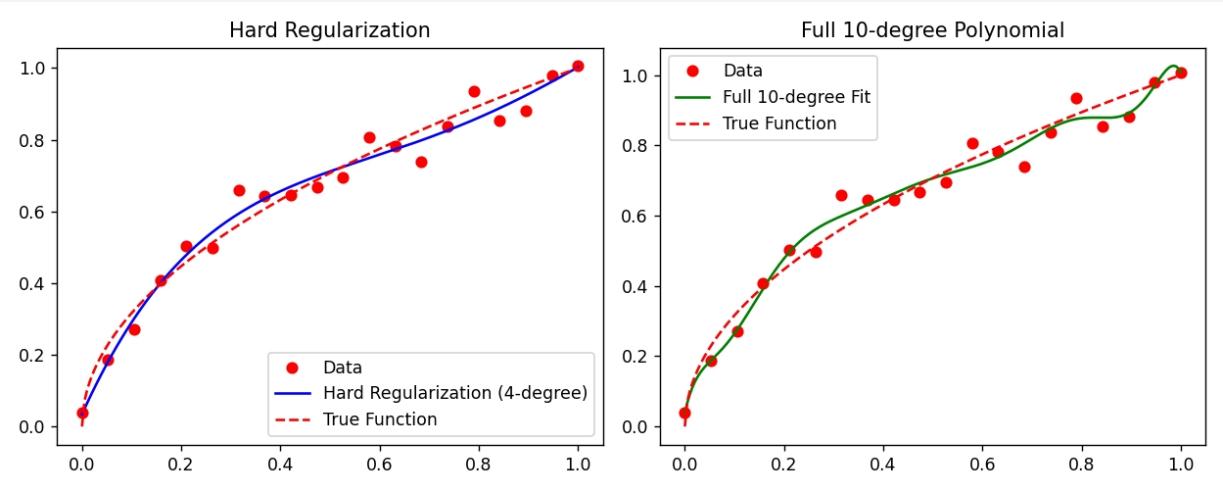
硬正则化的主要算法原理如下：

**限制模型结构**：硬正则化通过限制模型的结构来实现模型的简化。例如，可以限制卷积神经网络中卷积核的大小、通道数或卷积层的数量等。

**限制模型参数**：硬正则化通过限制模型的参数来实现模型的简化。例如，可以限制全连接层中权重的范围、限制L1正则化或L2正则化等。

限制模型结构代码





左图（硬正则化）减少特征数量，控制模型复杂度，明显偏向于简单模型，防止过拟合，可能会欠拟合

右图（10阶线性回归）训练数据拟合得好，但在边界处波动大，出现过拟合趋势

硬正则化的应用范围有限，主要适用于卷积神经网络和循环神经网络等结构，对于其他类型的模型，硬正则化的应用需要进一步探索