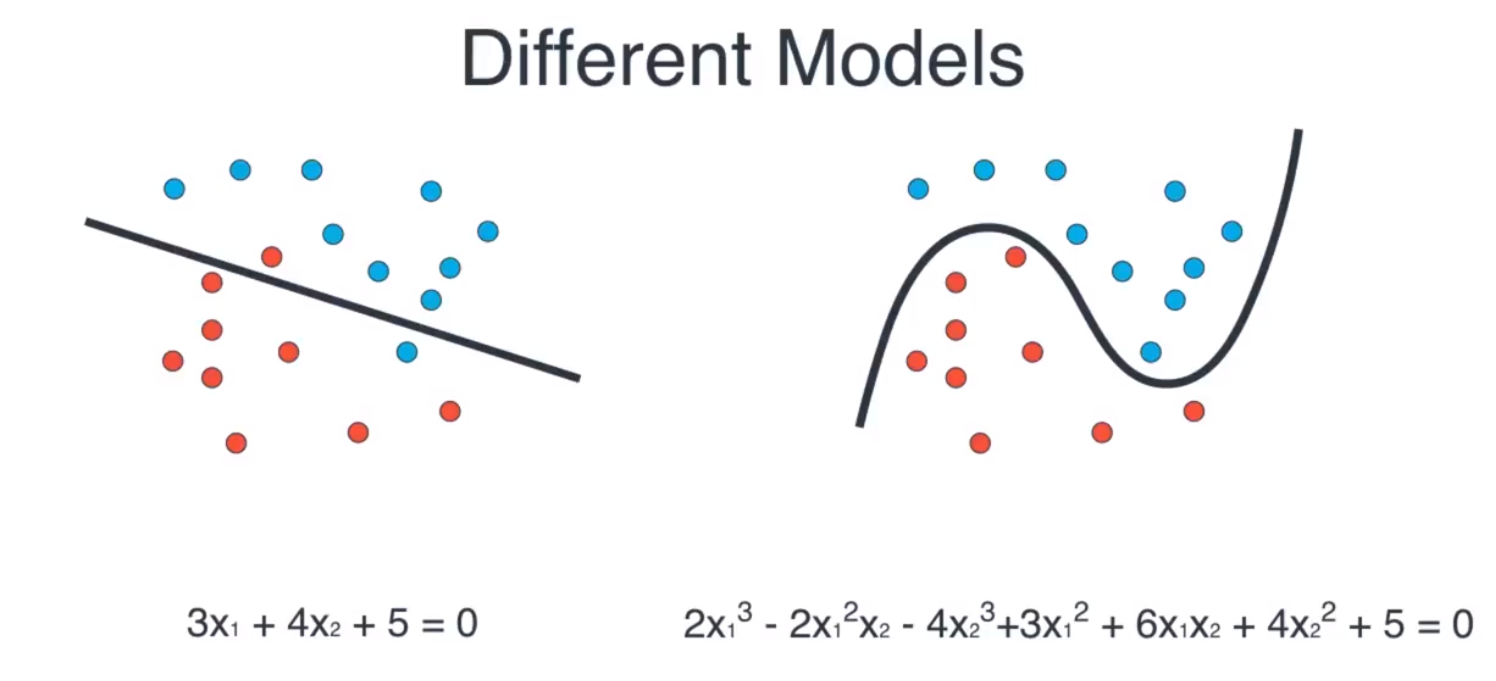
正则化：是一个非常重要的概念能帮助我们改善模型，是确保不会过度拟合的有效技巧，可用于回归和分类

实际上可以利用分类问题对它进行解释 但是正如你看到的，所有参数值都与回归算法一样发挥作用

模型

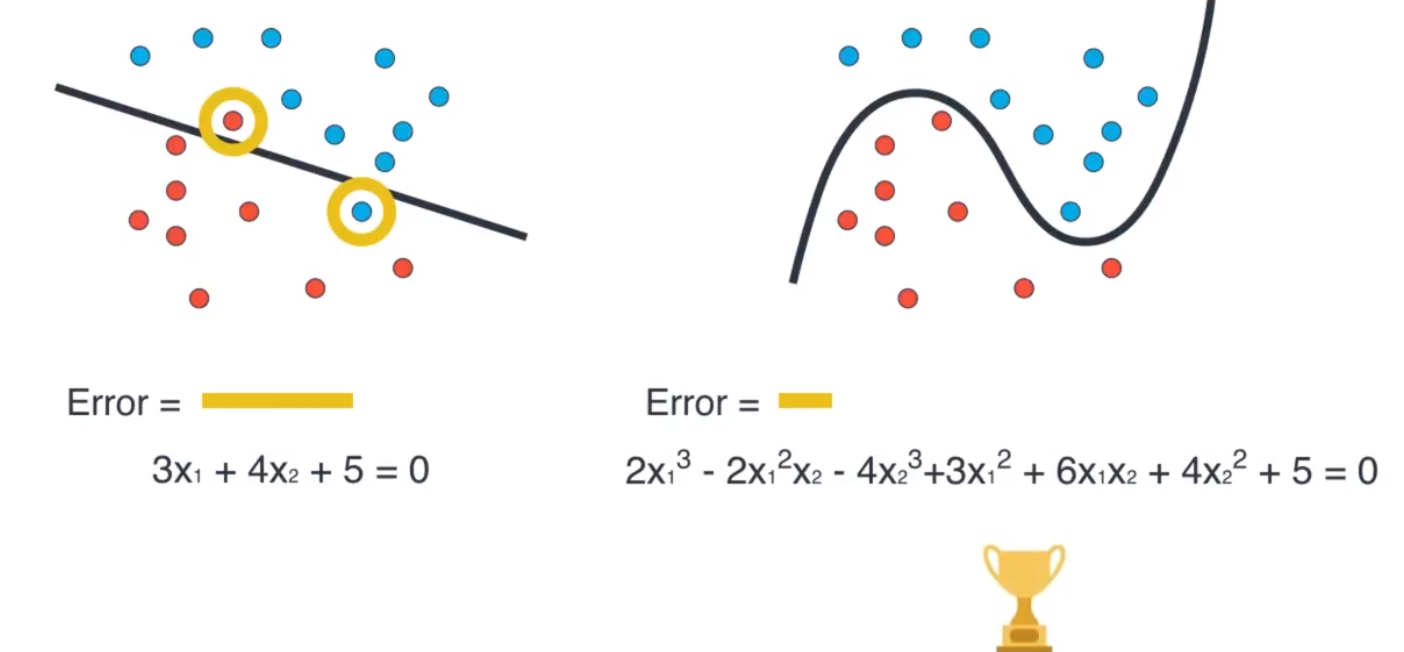


第一个是一条直线，第二个是高次多项式，那么问题是 哪个更好呢？

两种方法是不是各有利弊？

1.左侧这个存在各种错误，可以看到 有一个红点和蓝点都在错误位置，但是这种方法更加简单

2.右侧这个没有任何错误，但实际上有点更加复杂，那么如果我们想要左侧这个，因为右侧这个过度拟合，并且不具有较好的通用性

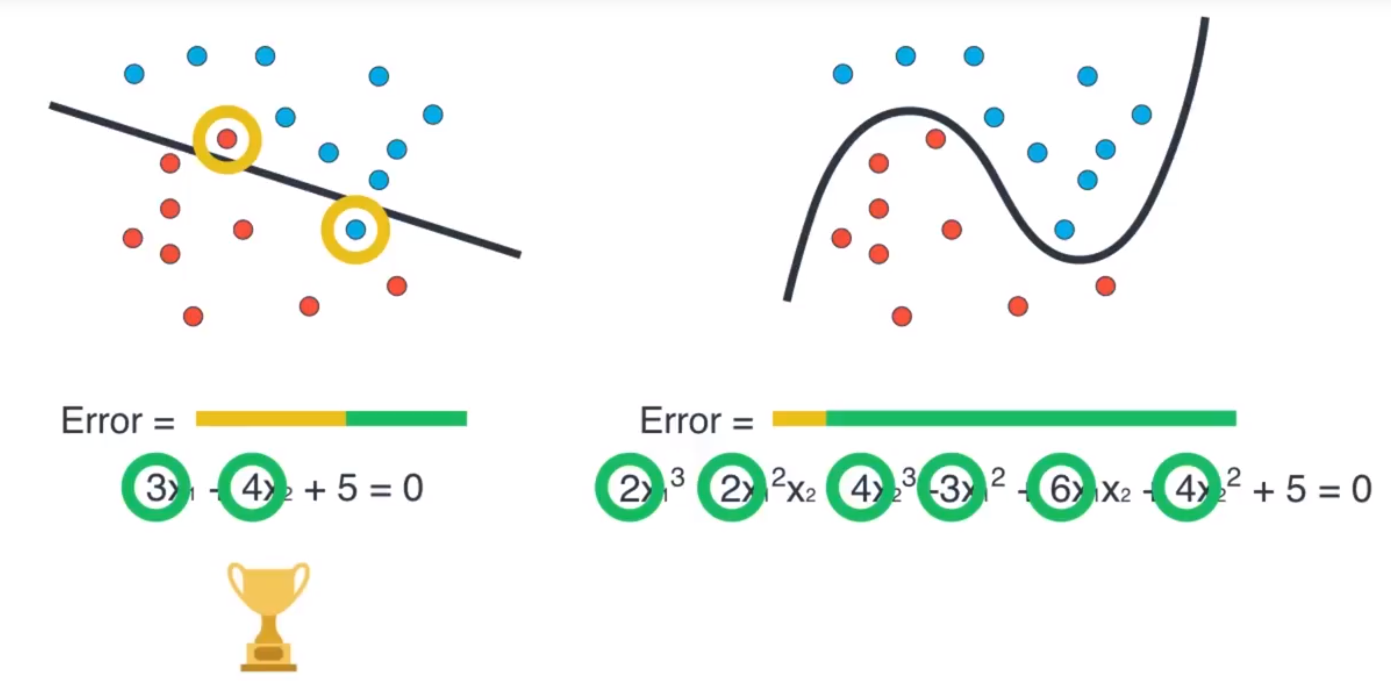


当我们训练模型时，得到模型误差，并进行最小化。

所以左侧这个误差较小，有两个错误，因此只有一很大的个误差

因为右侧这个分类没有错误，所以误差很小

从误差的角度选择 右侧模型



观察这个多项式，左侧多项式 系数 要比右侧 的更加简单，如果我们通过这些数字的函数找到增加误差的方法

注意我们这里忽略了常数项。

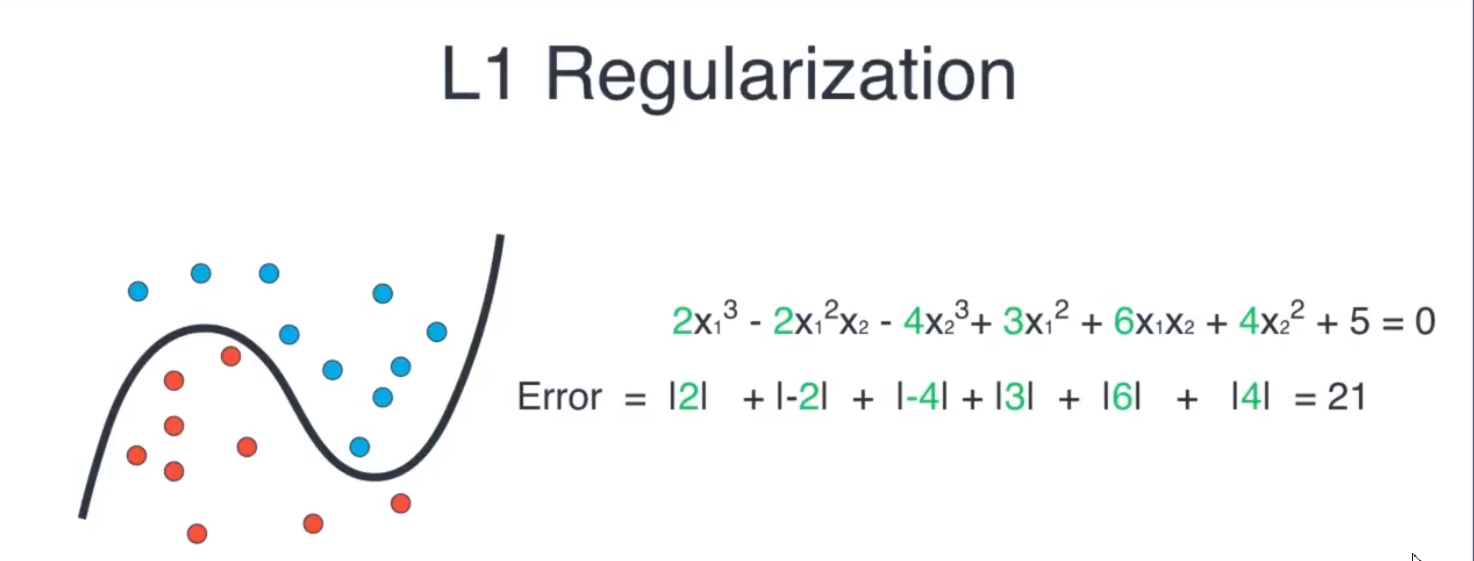
现在我们看到左侧模型更好一些，因为它的误差更小

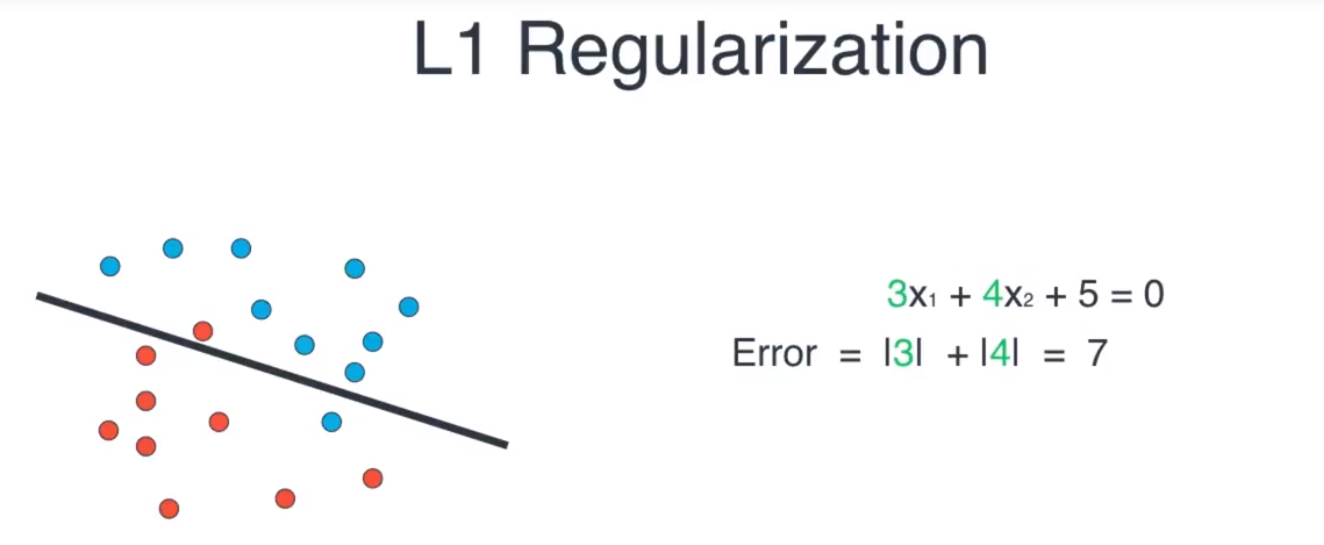
总结：

总之我们在计算误差时要考虑模型的复杂性，简单的模型比复杂的模型更胜一筹，简单的模型更具有通用性，那是我们想要的

L1正则化

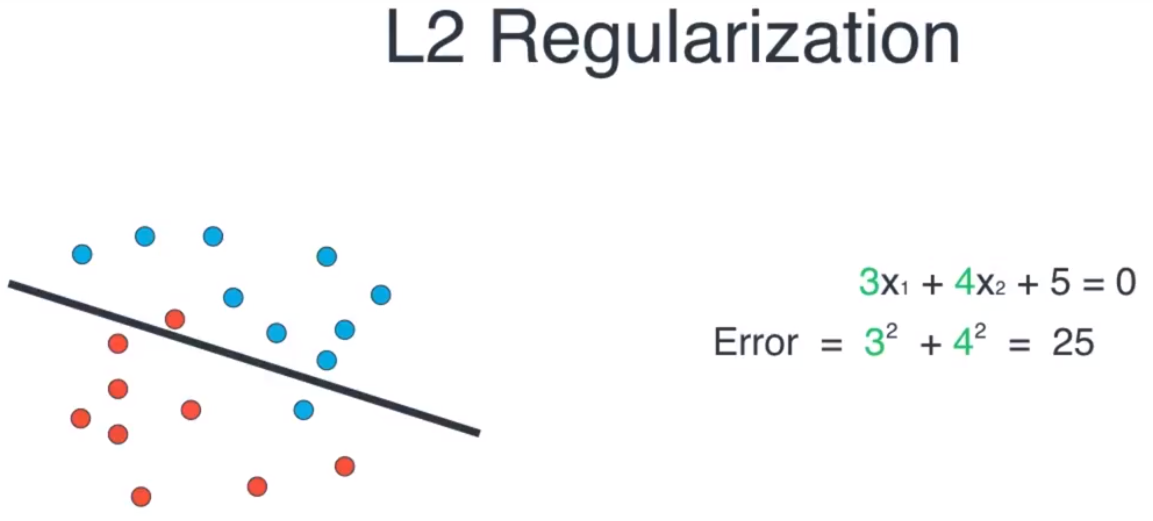
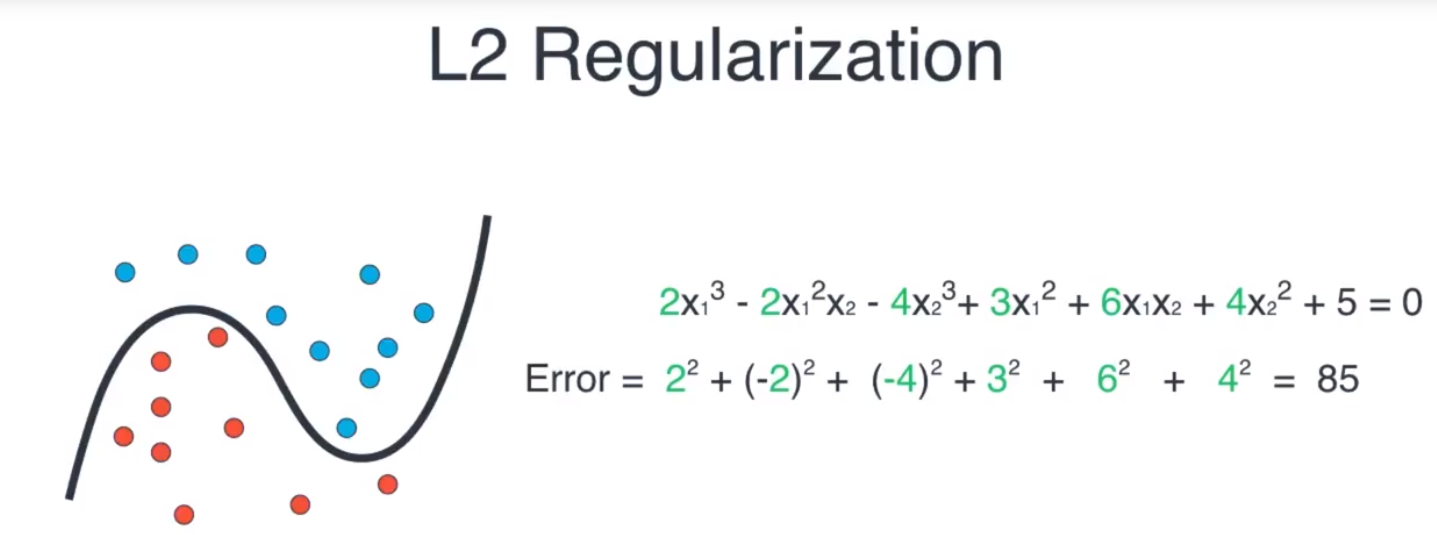
利用多项式的系数 把它们转化为误差的一部分，L1正则化在于把它们的绝对值加起来，这就是为什么复杂的模型，得出较大的误差，这就是 L1 正则化 它与绝对值相关。





L2正则化

我们这里要做的，不是得到绝对值，而是把系数的平方加起来。所以，复杂的模型比简单的模型受到更多惩罚



参数调优

我们如何调优这些参数，因为我们想用更复杂一点的模型，所以我们这个参数称为 λ。

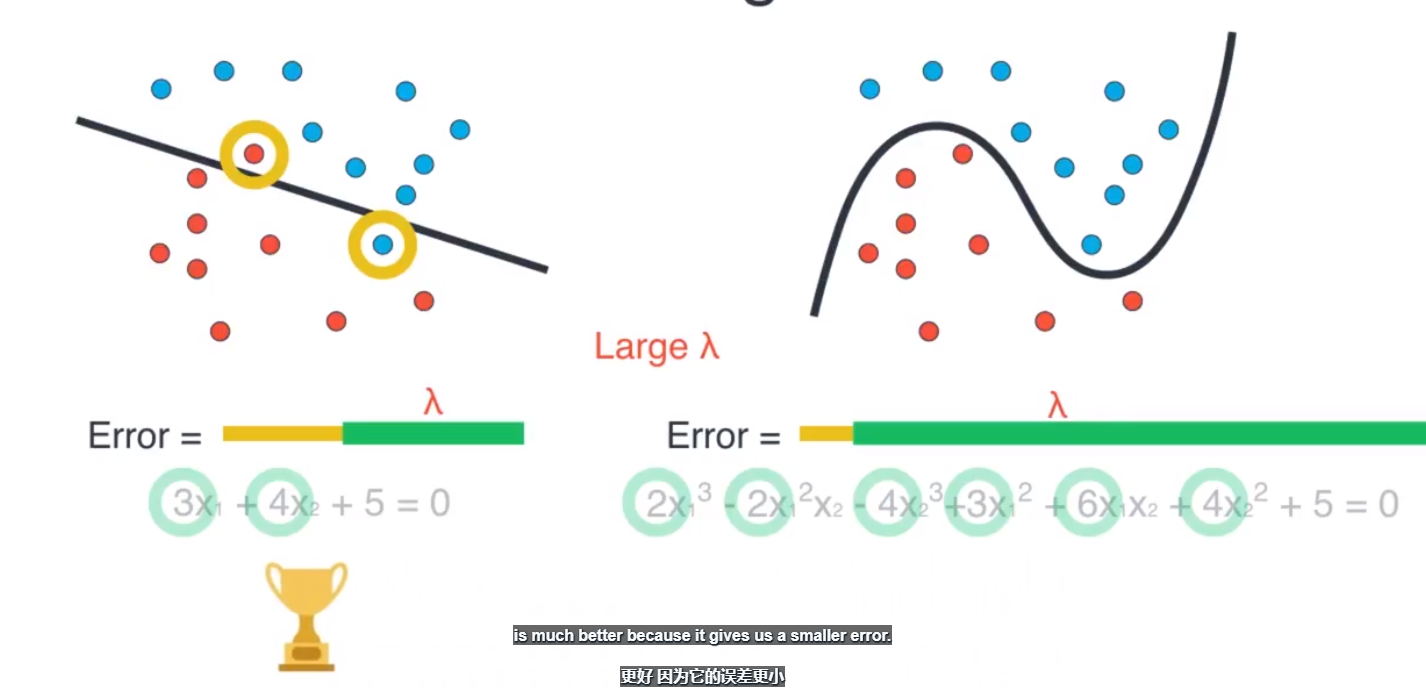
λ 的用途是可以乘以复杂误差（就是正则化L1,或L2），乘以那个绿色的误差可以变大或变小.

1. 如果有个小的 λ，我们得到绿色的误差 把它乘以 λ，我们就会得到很小的结果，因为误差更小，所以右侧还是胜出



我们得到某个复杂误差 但是这个误差更小，如果我们有个大的 λ 乘以绿色的误差

这是可以看到简单的模型更好 因为它的误差更小



总结

所以总体说来 如果我们有个大的 λ

我们会更多地惩罚复杂性，拾取更加简单的模型

如果我们有个小的 λ 我们会以较小数量惩罚复杂性，所以我们可以使用更复杂的模型。

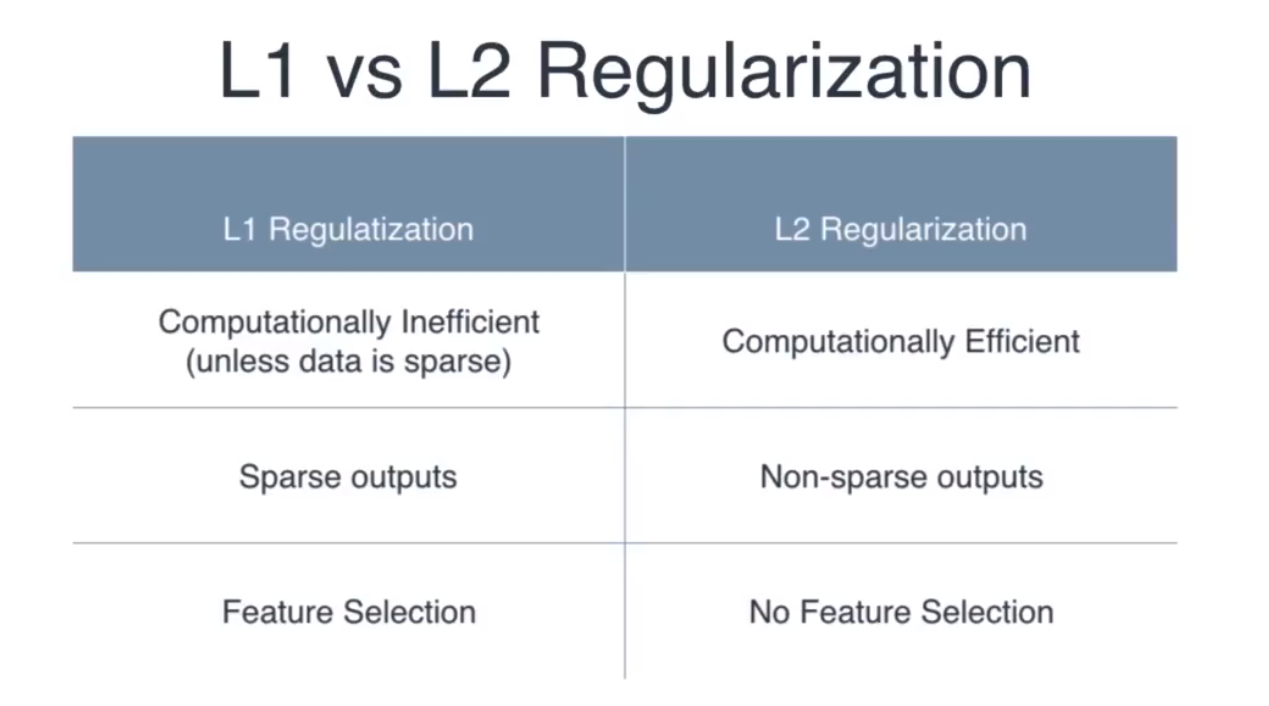
L1与L2比较

现在的问题是我们应该使用哪个呢？ L1 还是 L2？

L1 正则化实际上计算起来，效率不高 虽然看起来更简单，因为这些绝对值很难求导(求微分)

L2 正则化中，平方是比较不错的导数，所以可以计算起来更加容易

只有数据稀疏时 L1 正则化会比 L2 正则化更快



如果很多列包含 0 那么 L1 正则化更好些，但是如果数据均匀分布在各列 那么 L2 正则化更好些

L1 正则化最大的好处在于提供特征选择，

如果我们有 1000 列数据，实际上只有 10 列数据相关，其他的都是噪音

L1 可以注意到这一点 把不相关的数据转变为 0

L2 就无法做到这一点，L2 会对所有列平等对待

并且给我们所有数据的总和作为结果

这就是正则化