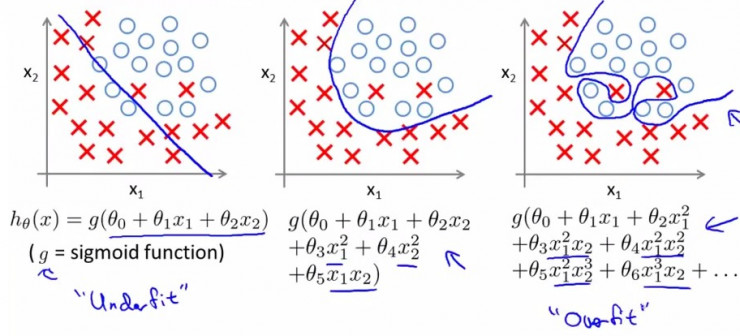
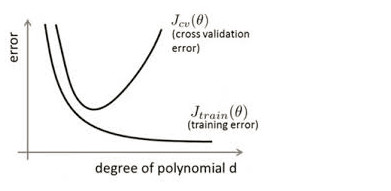
1. 什么是过拟合

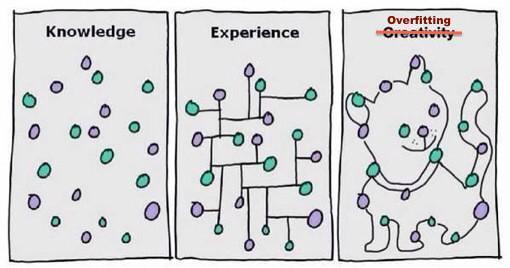
过拟合就是训练出来的模型在训练集上表现很好，但是在测试集上表现较差的一种现象！下图给出例子：



我们将上图第三个模型解释为出现了过拟合现象，过度的拟合了训练数据，而没有考虑到泛化能力。在训练集上的准确率和在开发集上的准确率画在一个图上如下：



从图中我们能够看出，模型在训练集上表现很好，但是在交叉验证集上表现先好后差。这也正是过拟合的特征



1. 过拟合的原因：
2. 训练集与测试集的特征分布不一致
3. 样本量不足，模型太过复杂，学习了太多的噪声

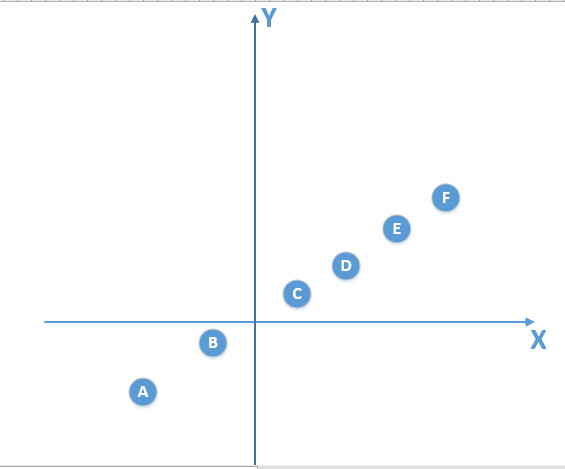
1）数据有噪声

为什么数据有噪声，就可能导致模型出现过拟合现象呢？

所有的机器学习过程都是一个 search 假设空间的过程！我们是在模型参数空间搜索一组参数，使得我们的损失函数最小，也就是不断的接近我们的真实假设模型，而真实模型只有知道了所有的数据分布，才能得到。

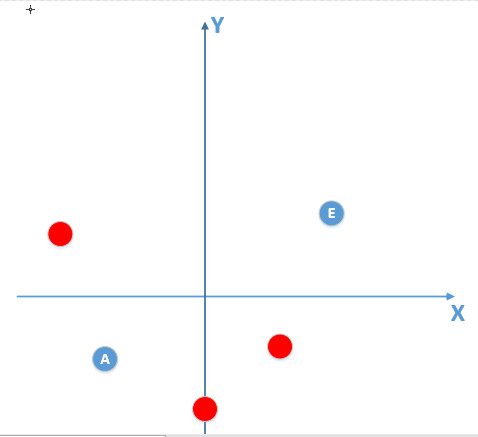
往往我们的模型是在训练数据有限的情况下，找出使损失函数最小的最优模型，然后将该模型泛化于所有数据的其它部分。这是机器学习的本质！

那好，假设我们的总体数据如下图所示：



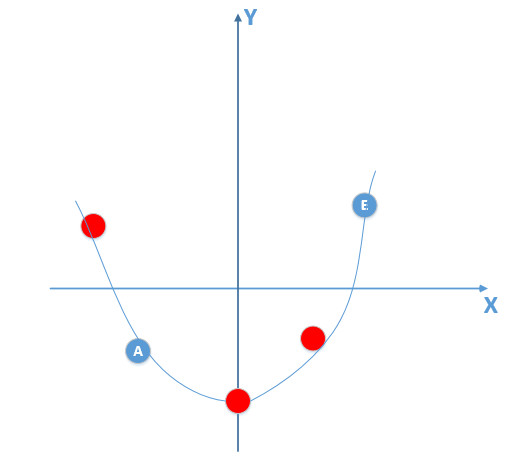
（我这里就假设总体数据分布满足一个线性模型 y = kx+b, 现实中肯定不会这么简单，数据量也不会这么少，至少也是多少亿级别，但是不影响解释。反正总体数据满足模型 y）

此时我们得到的部分数据，还有噪声的话，如图所示：



（红色数据点为噪声）

那么由上面训练数据点训练出来的模型肯定不是线性模型（总体数据分布下满足的标准模型），比如训练出来的模型如下：

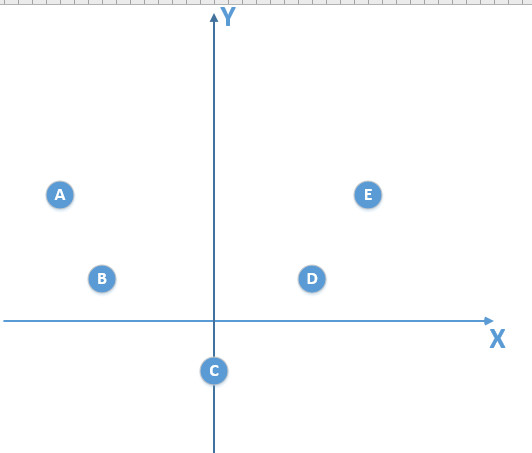


那么我拿着这个有噪声训练的模型，在训练集合上通过不断训练，可以做到损失函数值为 0，但是拿着这个模型，到真实总体数据分布中（满足线性模型）去泛化，效果会非常差，因为你拿着一个非线性模型去预测线性模型的真实分布，显而易得效果是非常差的，也就产生了过拟合现象！

（2）训练数据不足，有限的训练数据

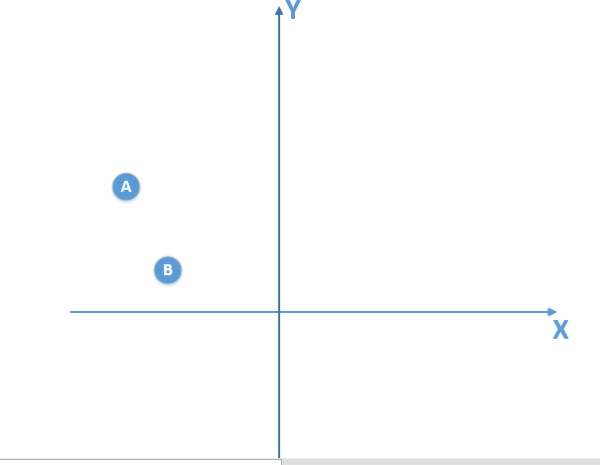
当我们训练数据不足的时候，即使得到的训练数据没有噪声，训练出来的模型也可能产生过拟合现象，解释如下：

假设我们的总体数据分布如下：



（为了容易理解，假设我们的总体数据分布满足的模型是一个二次函数模型）

我们得到的训练数据由于是有限的，比如是下面这个：



（我只得到了 A，B 两个训练数据）

那么由这个训练数据，我得到的模型是一个线性模型，通过训练较多的次数，我可以得到在训练数据使得损失函数为 0 的线性模型，拿这个模型我去泛化真实的总体分布数据（实际上是满足二次函数模型），很显然，泛化能力是非常差的，也就出现了过拟合现象！

（3）训练模型过度导致模型非常复杂

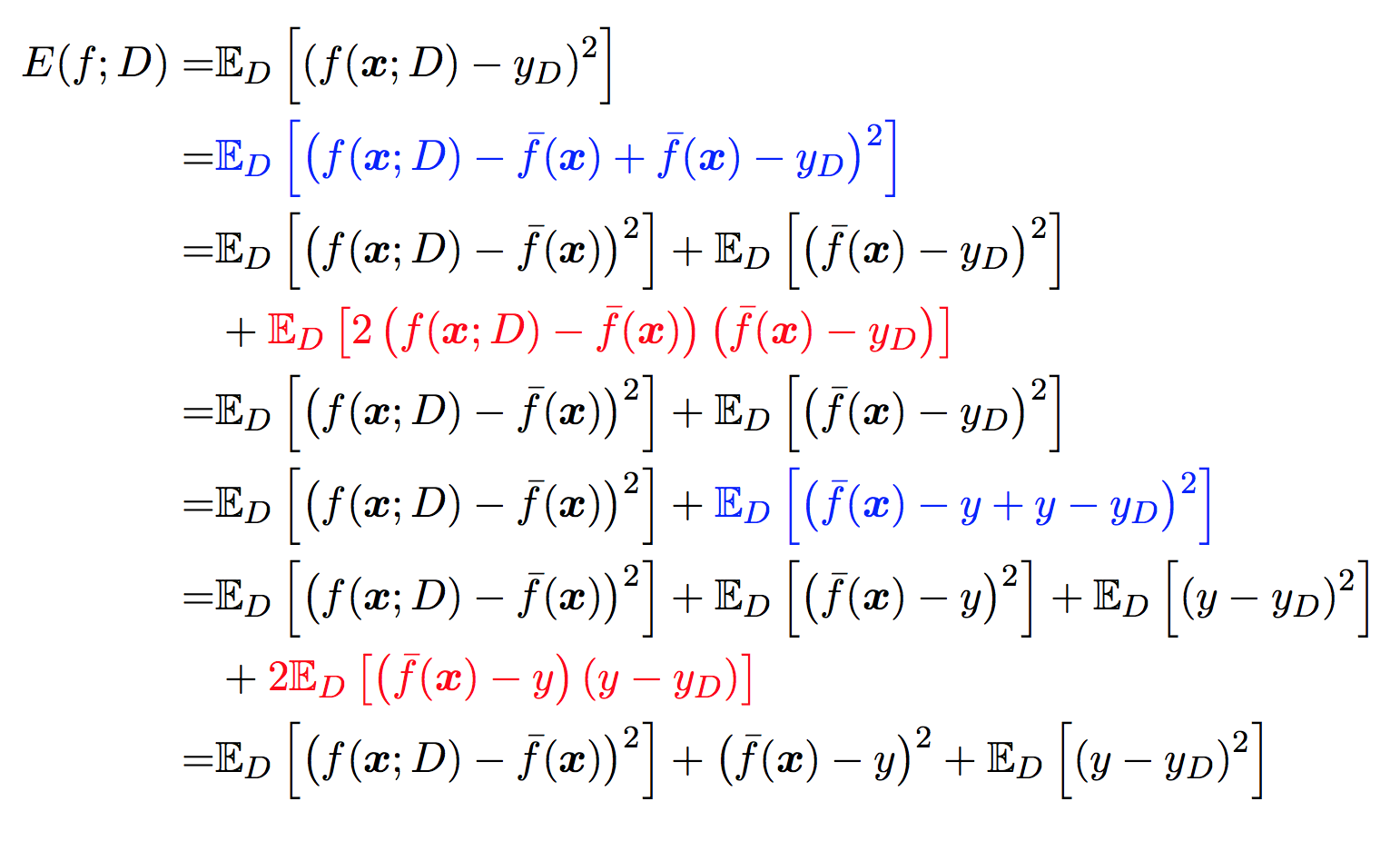
训练模型过度导致模型非常复杂，也会导致过拟合现象！这点和第一点俩点原因结合起来其实非常好理解，当我们在训练数据训练的时候，如果训练过度，导致完全拟合了训练数据的话，得到的模型不一定是可靠的。

比如说，在有噪声的训练数据中，我们要是训练过度，会让模型学习到噪声的特征，无疑是会造成在没有噪声的真实测试集上准确率下降！

1. 如何判断过拟合
2. 方差，偏差

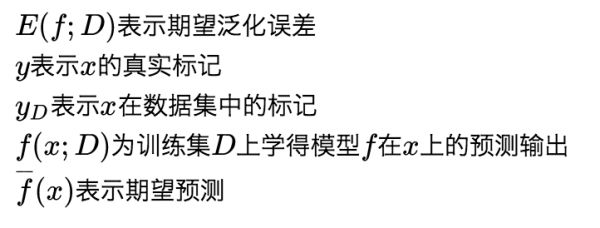
方差：方差衡量的是**模型自身**的**稳定性**，就是说模型每次输出都会有或大或小的抖动，方差反映的是f(x)与E[f(x)]之间的差别是否稳定

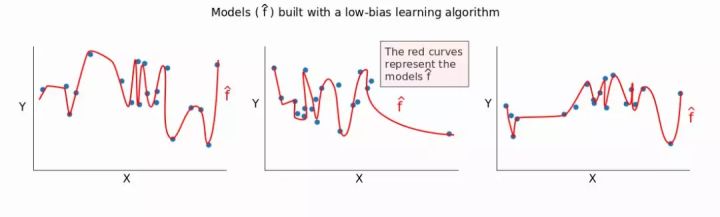
偏差：偏差反映的是**预测值**期望E[f(x)]与**真实值**y之间**偏离**了多少。



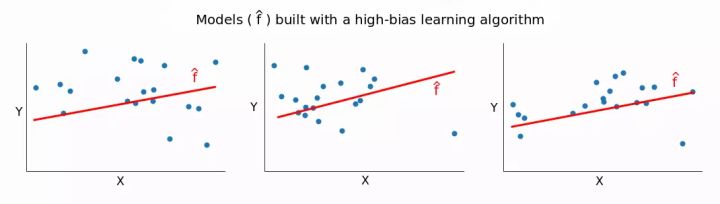
https://pic3.zhimg.com/80/v2-3fa78f3336daf3a1e661b1ab191972f9_hd.jpg

https://pic1.zhimg.com/80/v2-90d3827393a354739f8730c9de3892f5_hd.jpg

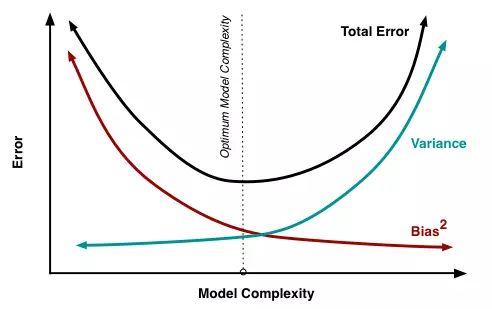


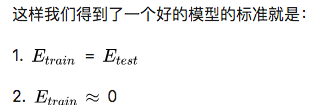


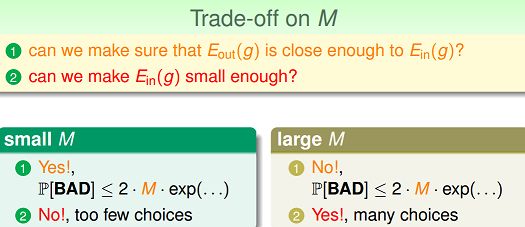
模型的 bias 越低，它适应数据的能力就越强，同时 variance 也越高。所以，bias 越低，variance 越高。

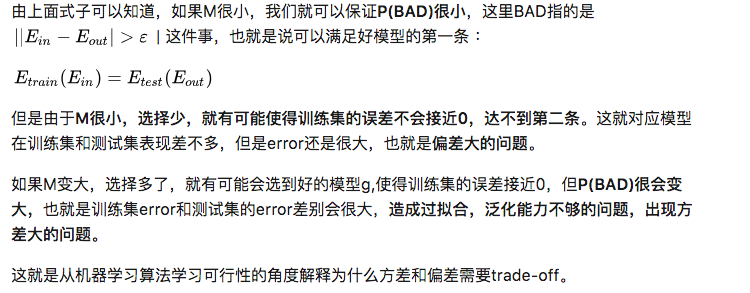


bias 越高，variance 越低。一个高 variance 的模型构建的简单模型通常是不能很好适应数据集的。当我们改变数据集的时候，从高 bias 的算法得到的模型 f^ 通常不会有很大不同。





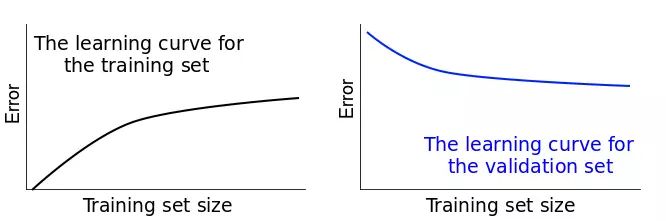




在实际中，我们需要接受一个 trade-off。我们不可能同时得到低 bias 和低 variance，所以我们期望得到某种中间结果。

2，学习曲线

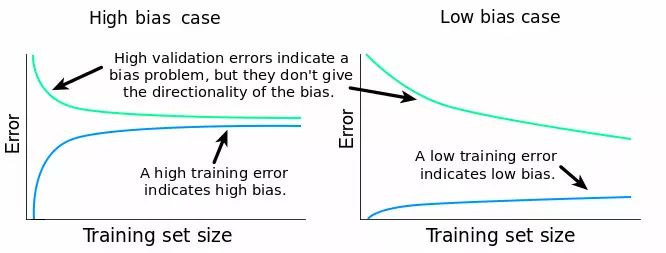
学习曲线会展示误差是如何随着训练集的大小的改变而发生变化的。



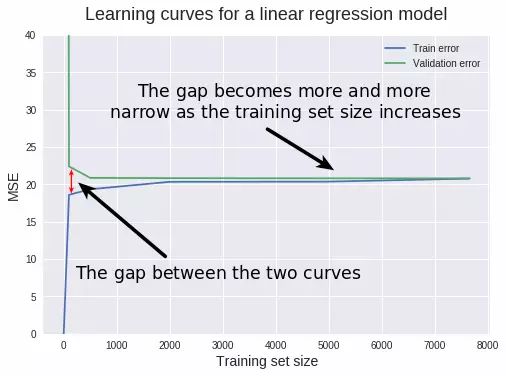
通过bias诊断

如果训练误差特别小，这就说明估计模型能够很好地拟合训练数据，这就是说模型在对应的数据集上有较小的 bias。

如果训练误差比较高，就说明估计模型不能很好地拟合训练数据，也就意味着在对应的数据集上有较高的 bias。



对 variance 的估计可以通过以下两种方式完成：

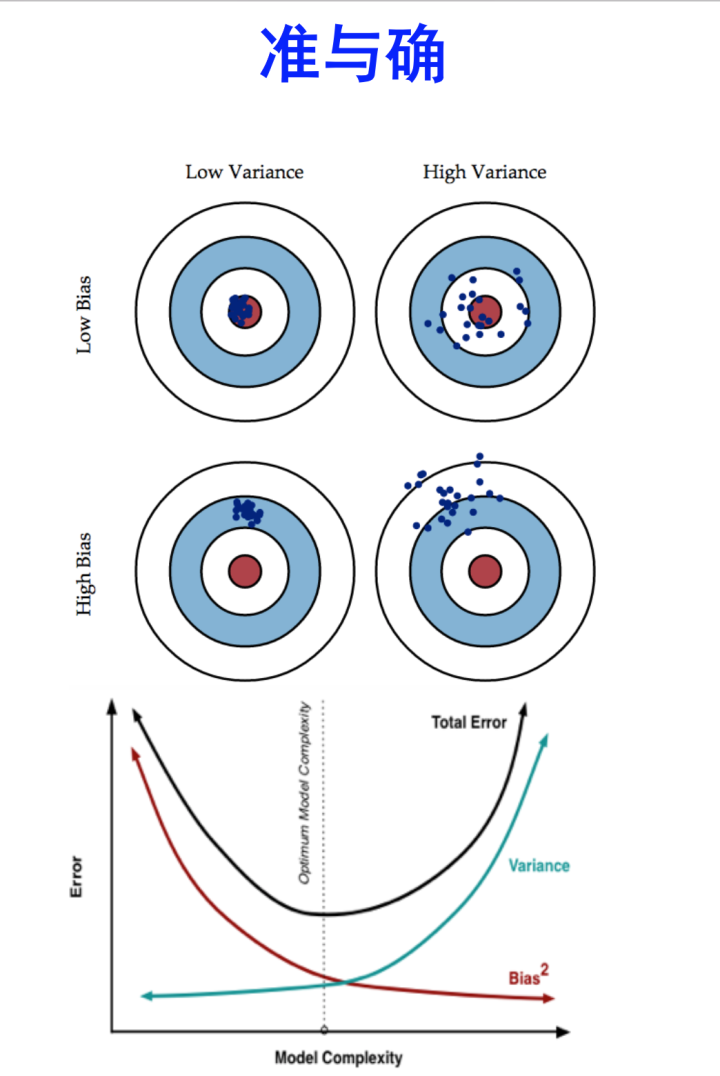
* 通过检查验证学习曲线和训练学习曲线之间的差距；
* 通过检查训练误差：检查误差的值随着训练样本数增加的变化。
* 

较小的gap差距代表较小的 variance。通常，差距越小，variance 越小。反之亦然：差距越大，variance 越大

目前，我们可以总结如下：

高 bias，低 variance，以及对训练数据的欠拟合 🡪换模型，尝试增加训练样本的数量是一个误区。

低bias 高variance，🡪增加数据，换模型，正则，early stopping,drop out



随机森林：高bias，低variance

gbdt：低bias 高variance

1. 防止过拟合：
2. 增加数据：

从数据源头采集更多数据

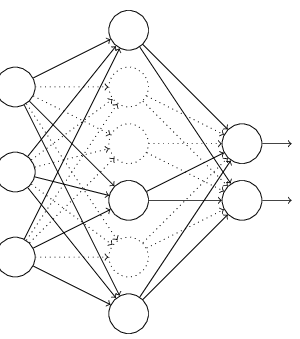
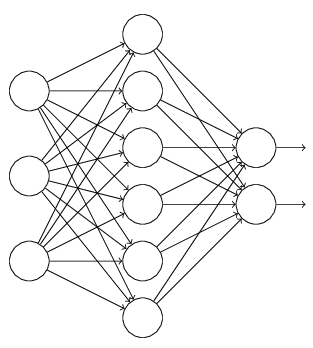
复制原有数据并加上随机噪声

重采样

根据当前数据集估计数据分布参数，使用该分布产生更多数据等

2，数据不纯，含有大量的噪声，要清洗数据

3，drop out



4，正则

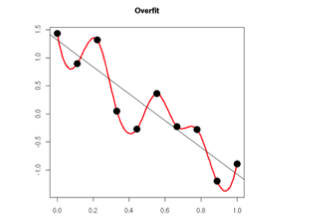
L0:向量非0元素的个数，是NP难问题

L1:向量元素绝对值的和，是L0的近似

L2:向量元素平方和再开根号

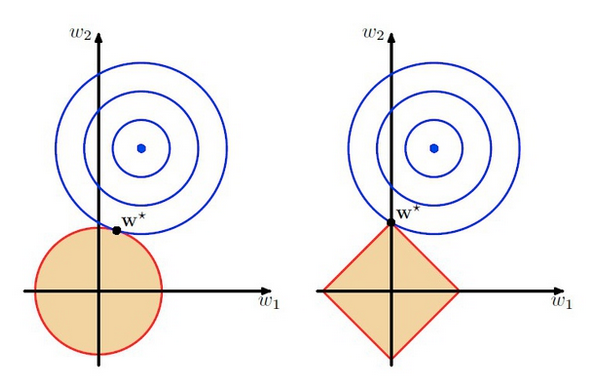
正则项是为了降低模型的复杂度，从而避免模型区过分拟合训练数据，包括噪声与异常点（outliers）。从另一个角度上来讲，正则化即是假设模型参数服从先验概率，即为模型参数添加先验，只是不同的正则化方式的先验分布是不一样的。这样就规定了参数的分布，使得模型的复杂度降低（试想一下，限定条件多了，是不是模型的复杂度降低了呢），这样模型对于噪声与异常点的抗干扰性的能力增强，从而提高模型的泛化能力。还有个解释便是，从贝叶斯学派来看：加了先验，在数据少的时候，先验知识可以防止过拟合；从频率学派来看：正则项限定了参数的取值，从而提高了模型的稳定性，而稳定性强的模型不会过拟合，即控制模型空间。

另外一个角度，过拟合从直观上理解便是，在对训练数据进行拟合时，需要照顾到每个点，从而使得拟合函数波动性非常大，即方差大。在某些小区间里，函数值的变化性很剧烈，意味着函数在某些小区间里的导数值的绝对值非常大，由于自变量的值在给定的训练数据集中的一定的，因此只有系数足够大，才能保证导数的绝对值足够大。



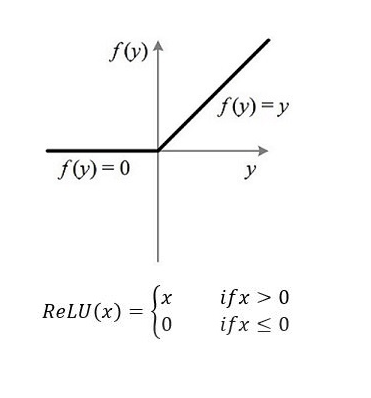
另外一个解释，规则化项的引入，在训练（最小化cost）的过程中，当某一维的特征所对应的权重过大时，而此时模型的预测和真实数据之间距离很小，通过规则化项就可以使整体的cost取较大的值，从而，在训练的过程中避免了去选择那些某一维（或几维）特征的权重过大的情况，即过分依赖某一维（或几维）的特征

L2与L1的区别在于，L1正则是拉普拉斯先验，而L2正则则是高斯先验。它们都是服从均值为0，协方差为1/a时，当a=0时（即没有先验）没有正则项，则相当于先验分布具有无穷大的协方差，那么这个先验约束则会非常弱，模型为了拟合所有的训练集数据， 参数w可以变得任意大从而使得模型不稳定，即方差大而偏差小。a越大，标明先验分布协方差越小，偏差越大，模型越稳定。即，加入正则项是在偏差bias与方差variance之间做平衡tradeoff。下图即为L2与L1正则的区别：



上图中的模型是线性回归，有两个特征，要优化的参数分别是w1和w2，左图的正则化是L2，右图是L1。蓝色线就是优化过程中遇到的等高线，一圈代表一个目标函数值，圆心就是样本观测值（假设一个样本），半径就是误差值，受限条件就是红色边界（就是正则化那部分），二者相交处，才是最优参数。可见右边的最优参数只可能在坐标轴上，所以就会出现0权重参数，使得模型稀疏。

其他方法：relu， batch normal等，可以减少参数的，可以使参数波动小的，使数据纯，全的方法都能防止过拟合。



https://blog.csdn.net/heyongluoyao8/article/details/49429629

https://zhuanlan.zhihu.com/p/33220323

https://zhuanlan.zhihu.com/p/28673680