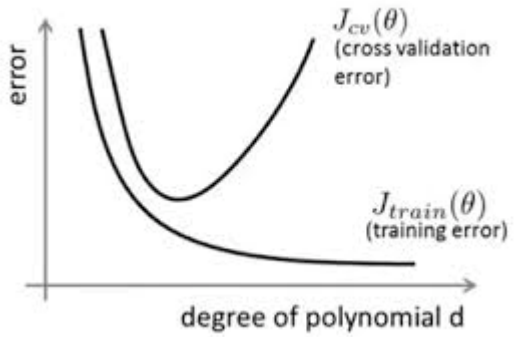
机器学习中常见的过拟合解决方法

在机器学习中，我们将模型在训练集上的误差称之为训练误差，又称之为经验误差，在新的数据集（比如测试集）上的误差称之为泛化误差，泛化误差也可以说是在模型在总体样本上的误差。对于一个好的模型应该是经验误差约等于泛化误差，也就是经验误差要收敛于泛化误差，根据霍夫丁不等式可知经验误差在一定条件下是可以收敛于泛化误差的。

当机器学习模型对训练集学习的太好的时候（再学习数据集的通性的时候，也学习了数据集上的特性，这些特性是会影响模型在新的数据集上的表达能力的，也就是泛化能力），此时表现为经验误差很小，当往往此时的泛化误差会很大，这种情况我们称之为过拟合，而当模型在数据集上学习的不够好的时候，此时经验误差较大，这种情况我们称之为欠拟合。具体表现如下图所示，第一幅图就是欠拟合，第三幅图就是过拟合。

再如下面这幅图所示，只要我们愿意，在训练集上的误差是可以无限小的，但是此时的泛化误差会增大。



对于欠拟合的问题比较好处理，只要增大模型的复杂度就行，而且欠拟合会在训练过程中就表现出来，更容易去控制。对于过拟合不会体现在训练集上，因此常见的方式是采用交叉验证也检测过拟合。解决过拟合的两条主线：一是增大数据集，而是降低模型的复杂度（根据VC维理论可知）。一般来说扩展数据集是比较难的，而且数据集大，模型复杂度高的时候即使能获得好的泛化结果，也会增大计算量。所以常见的方式都是以降低模型的复杂度为主，接下来看看有哪些常见的方法可以降低模型的复杂度

1、正则化

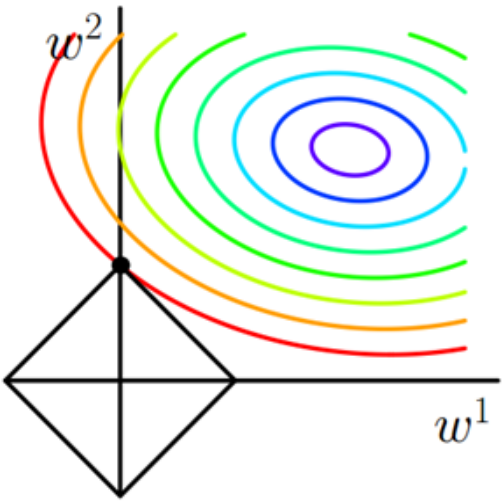
正则化是机器学习中最常见的过拟合解决方法，在损失函数中加入正则项来惩罚模型的参数，以此来降低模型的复杂度，常见的正则化技术有L1，L2正则化。

L1正则化

$$J = J_0 + \alpha \sum_w |w|$$

L1正则化是基于L1范数的，J是我们的损失函数，在损失函数优化时，我们要使得损失函数无限小，要满足这个结果，表达式中的第二项也必须无限小。关于L1正则化项的函数可以在二维平面图中表示出来，令

$$L = \alpha \sum_w |w|$$

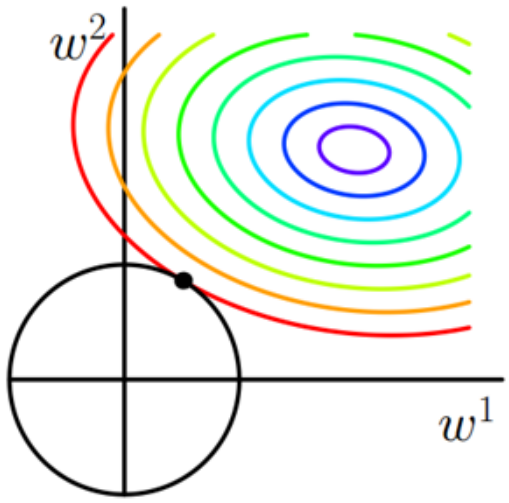


图中的等值线（就是那些曲线）是J_0的等值线，黑色方形是正则项函数L的图形。在图中。当J_0等值线和L图形首次相交的地方就是最优解（根据拉格朗日约束问题的最小值可知，L函数可以看作J_0函数的约束，没有该约束，J_0函数的最小值应该是最里面的等值线，加上约束之后，就是首次相交的点处），从图中可以看出在相交点这里有w_1为0，这也是L1正则的特点。加入L1正则项之后，数据集中那些对模型贡献不大的特征所对应的参数w可以为0，因此L1正则项得出的参数是稀疏的。

L2正则化

$$J = J_0 + \alpha \sum_w w^2$$

同样可以画出在二维平面中的图形来描述



原理和L1正则中差不多，但是L2正则化不会获得稀疏解，只会将对模型贡献不大的特征所对应的参数置于无限小的值，以此来忽略该特征对模型的影响。

因此正则化都是在通过控制模型参数的大小来降低模型的复杂度（VC维原理）

2、剪枝处理

剪枝是决策树中一种控制过拟合的方法，我们知道决策树是一种非常容易陷入过拟合的算法，剪枝处理主要有预剪枝和后剪枝这两种，常见的是两种方法一起使用。预剪枝通过在训练过程中控制树深、叶子节点数、叶子节点中样本的个数等来控制树的复杂度。后剪枝则是在训练好树模型之后，采用交叉验证的方式进行剪枝以找到最优的树模型。

3、提前终止迭代（Early stopping）

该方法主要是用在神经网络中的，在神经网络的训练过程中我们会初始化一组较小的权值参数，此时模型的拟合能力较弱，通过迭代训练来提高模型的拟合能力，随着迭代次数的增大，部分的权值也会不断的增大。如果我们提前终止迭代可以有效的控制权值参数的大小，从而降低模型的复杂度。

4、权值共享

权值共享最常见的就是在卷积神经网络中，权值共享的目的旨在减小模型中的参数，同时还能较少计算量。在循环神经网络中也用到了权值共享。

5、增加噪声

这也是深度学习中的一种避免过拟合的方法（没办法，深度学习模型太复杂，容易过拟合），对于这种方法不甚了解。

6、Batch Normalization

BM算法是一种非常有用的正则化方法，而且可以让大型的卷积神经网络快速收敛，同时还能提高分类的准确率，而且可以不需要使用局部响应归一化处理，也可以不需要加入Dropout。BM算法会将每一层的输入值做归一化处理，并且会重构归一化处理之后的数据，确保数据的分布不会发生变化。

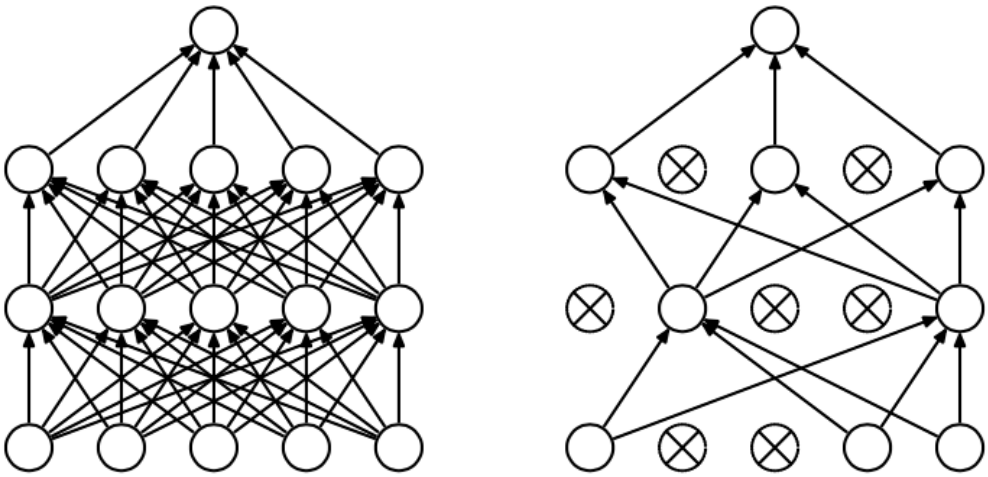
上面的几种方法都是操作在一个模型上 ，通过改变模型的复杂度来控制过拟合。另一种可行的方法是结合多种模型来控制过拟合。

7、Bagging和Boosting

Bagging和Boosting是机器学习中的集成方法，多个模型的组合可以弱化每个模型中的异常点的影响，保留模型之间的通性，弱化单个模型的特性。

8、Dropout

Dropout是深度学习中最常用的控制过拟合的方法，主要用在全连接层处。Dropout方法是在一定的概率上（通常设置为0.5，原因是此时随机生成的网络结构最多）隐式的去除网络中的神经元，具体如下图



Dropout控制过拟合的思想和机器学习中的集成方法类似，在每次迭代时dropout的神经元都不一样，因此对于整个模型参数而言，每次都会有一些参数不被训练到。Dropout会导致网络的训练速度慢2、3倍，而且数据小的时候，Dropout的效果并不会太好。因此只会在大型网络上使用。