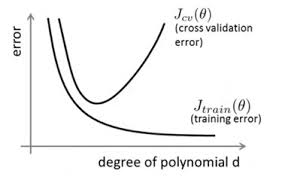
**关于过拟合**

给定一个假设空间H，一个假设h属于H，如果存在其他的假设h’属于H,使得在训练样例上h的错误率比h’小，但在整个实例分布上h’比h的错误率小，那么就说假设h过度拟合训练数据

在训练数据不够多时，或者overtraining时，常常会导致overfitting（过拟合）。其直观的表现如下图所示，随着训练过程的进行，模型复杂度增加，在training data上的error渐渐减小，但是在验证集上的error却反而渐渐增大——因为训练出来的网络过拟合了训练集，对训练集外的数据却不work。



在机器学习算法中，我们常常将原始数据集分为三部分：

1. training data

2. validation data

3. testing data。

**validation data是什么？**

它其实就是用来避免过拟合的，在训练过程中，我们通常用它来确定一些超参数（比如根据validation data上的accuracy来确定early stopping的epoch大小、根据validation data确定learning rate等等）。

**那为啥不直接在testing data上做这些呢？**

因为如果在testing data做这些，那么随着训练的进行，我们的网络实际上就是在一点一点地overfitting我们的testing data，导致最后得到的testing accuracy没有任何参考意义。因此，training data的作用是计算梯度更新权重，validation data如上所述，testing data则给出一个accuracy以判断网络的好坏。

**过拟合原因**

**（1）建模样本抽取错误**，

包括（但不限于）样本数量太少，抽样方法错误，抽样时没有足够正确考虑业务场景或业务特点，等等导致抽出的样本数据不能有效足够代表业务逻辑或业务场景；   
**（2）样本里的噪音数据干扰过大**，

大到模型过分记住了噪音特征，反而忽略了真实的输入输出间的关系；   
**（3）建模时的“逻辑假设”到了模型应用时已经不能成立了**。

任何预测模型都是在假设的基础上才可以搭建和应用的，常用的假设包括：假设历史数据可以推测未来，假设业务环节没有发生显著变化，假设建模数据与后来的应用数据是相似的，等等。如果上述假设违反了业务场景的话，根据这些假设搭建的模型当然是无法有效应用的。   
**（4）参数太多、模型复杂度高**

**（5）决策树模型**。

如果我们对于决策树的生长没有合理的限制和修剪的话，决策树的自由生长有可能每片叶子里只包含单纯的事件数据(event)或非事件数据（no event），可以想象，这种决策树当然可以完美匹配（拟合）训练数据，但是一旦应用到新的业务真实数据时，效果是一塌糊涂。   
**（6）神经网络模型。**

a.由于对样本数据,可能存在隐单元的表示不唯一,即产生的分类的决策面不唯一.随着学习的进行, BP算法使权值可能收敛过于复杂的决策面,并至极致.   
 b.权值学习迭代次数足够多(Overtraining),拟合了训练数据中的噪声和训练样例中没有代表性的特征.

**避免过拟合的方法**有很多：

1. early stopping
2. 正则化（Regularization）包括L1、L2（L2 regularization也叫weight decay）
3. dropout
4. **验证数据**
5. **交叉验证**
6. 模型融合本质上也是一种提高泛化能力的方法
7. 数据集扩增（Data augmentation）

**验证数据**

一个最成功的方法是在训练数据外再为算法提供一套验证数据,应该使用在验证集合上产生最小误差的迭代次数,不是总能明显地确定验证集合何时达到最小误差

**交叉验证方法**

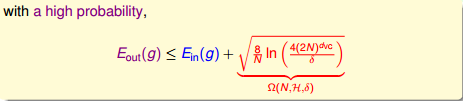
在可获得额外的数据提供验证集合时工作得很好,但是小训练集合的过度拟合问题更为严重.

**Early stopping**

所谓early stopping，即在每一个epoch结束时（一个epoch即对所有训练数据的一轮遍历）计算 validation data的accuracy，当accuracy不再提高时，就停止训练。这是很自然的做法，因为accuracy不再提高了，训练下去也没用。另外，这样做还能防止overfitting

**关于模型复杂度**

我们最终希望E-out越小越好，所以上面的不等式中可以只关心上界。我们把根号项看做一种惩罚，它拉大了E-in与E-out之间的距离，这个惩罚与“模型复杂度”有关，模型越复杂，惩罚越大：



上面的”模型复杂度“ 的惩罚(penalty)，基本表达了模型越复杂（VC维大），Eout 可能距离Ein 越远。下面的曲线可以更直观地表示这一点：一图胜千言，可以看出随着模型复杂度的增加，E-in与E-out两条曲线渐行渐远

