从VC维角度理解正则化与偏差方差权衡

正则化与偏差方差权衡是机器学习中的两个核心问题。本文尝试从VC维的角度阐释正则化与偏差方差权衡，以便更好的应用机器学习的方法。

本文主要包括以下四方面内容：

1、机器学习的框架

2、VC维理论（不做严格的证明，只说明VC维的概念，以及说明为什么VC维 可以作为模型复杂度的度量，以及论述偏差-方差权衡）

3、从VC维的角度理解正则化

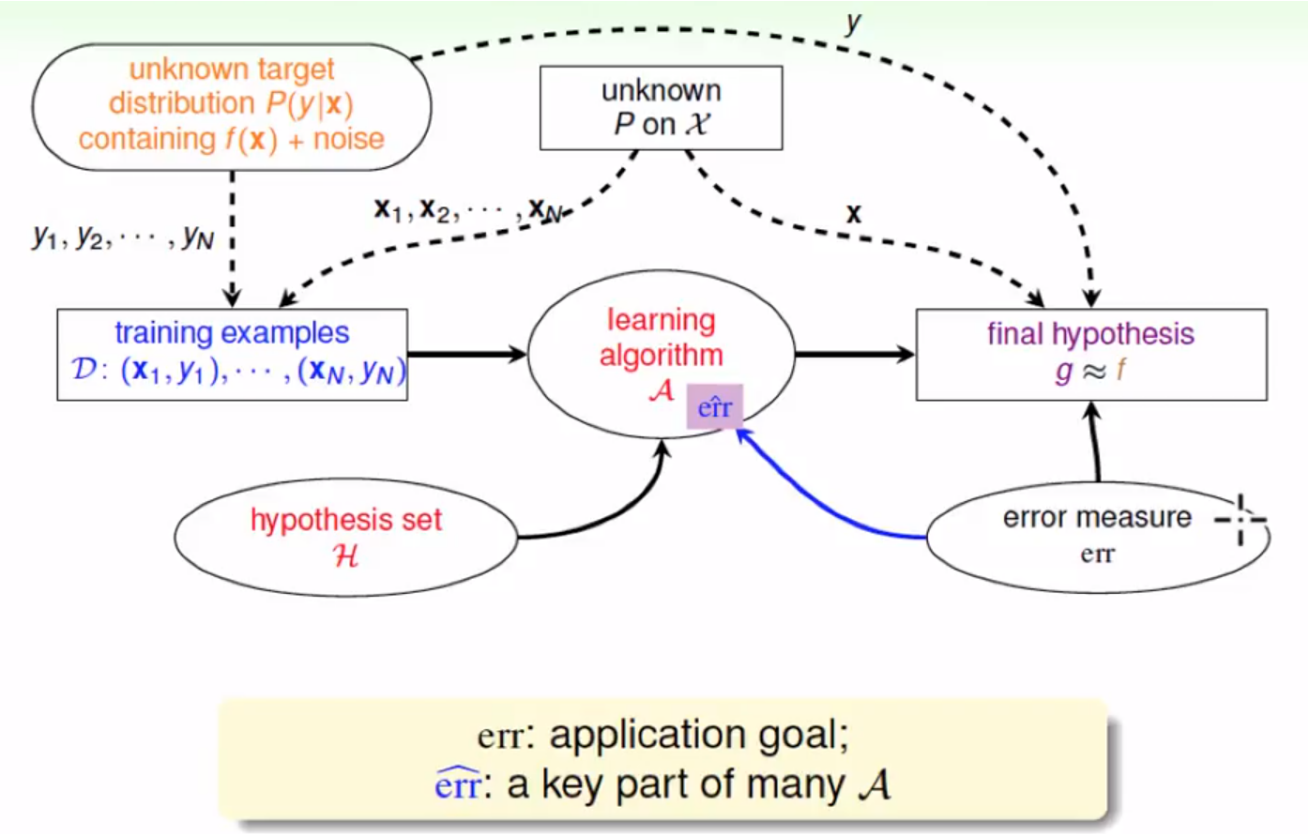
4、深度学习与VC维

5、深度学习中正则化与VC维解释

机器学习的框架

本质上说，机器学习是一种归纳推理的方法，即从一组有限的样本中推断一般的规则，因此，逻辑上无法保证在已经看过的数据之外一定能学到什么。事实上，对于我们没有观察到的数据，如果我们假设所有产生数据的规则f出现的概率相同，那么任何一种算法在没有观察到的数据上都有相同的错误率。这就是**天下没有免费午餐定理（No Free Lunch Theory)** 。但是NFL是在假设所有可能的规则f出现的概率相同时出现时才会成立，真实世界中，对于特定的问题，可以假设存在一些特定的**潜在的分布**（ X的产生）和**潜在的规则f**（X-->Y）,那么我们就可以设计算法进行学习，从而使得在我们关注的大多数样本上学到可能正确的规则（**PAC（Probably Approximately Correct)**)。

因此，机器学习的流程可以总结如下图所示：



其中，χ为产生X的潜在分布（假设训练集和测试集中的X都来源于同一分布χ，且相互独立），f为X-->Y的规则（由于噪音的存在，事实上我们得到的Y并不是精确的X通过规则f转化而得到的，因此我们采用概率分布P(Y|X)来描述训练集D的产生过程），A为机器学习的算法，H为假设空间（我们假设的所有可能的规则的集合），err为误差衡量的方式(有时err可能无法很好的优化，因此常采用err的替代函数$\hat{err}$)，g为最终学到的规则（函数）。

具体学习流程过程如下：

机器学习的算法A从假设的所有可能的规则的集合（假设空间H）中，选择能使对训练集上的数据(D)，对于给定的误差衡量方式$\hat{err}$，，误差最小的规则g作为最终的输出。

（有时我们通过在$\hat{err}$加入一些**先验知识**（比如我们希望模型尽可能比较平滑），这些先验知识无法从数据中学习得到，代表了我们的某些期望，事实上这就是**正则化**）。

但是，机器学习的目标并不是在我们观察到的训练集上获得最小的误差（训练误差），而是希望在我们尚未观察到的测试集上也有很好的表现，误差也足够小(泛化误差足够小，即有很好的**泛化能力(generalization)**)。

因此，机器学习可以归结为以下两个问题：

1、使得训练误差足够小

2、使得泛化误差与训练误差足够接近

对于第一个问题，选择合适的算法，给定足够大的假设空间，即可使得训练误差足够小（事实上神经网络的高度**可扩展性**（模型可以任意复杂）和**将优化与特征提取工作同时训练**的特性，可以使得我们可以获得任意小的训练误差，模型太过简单或者提取的特征本身都可能限制训练误差足够小）。

重要的是第二个问题，因此本文接下来尝试简单论述VC理论（以二分类问题为例，其他可以进行推广），从而说明在什么情况下，可以保证训练误差和泛化误差足够接近，保证机器可以学到东西。

VC 维

有限假设空间内的学习保证：

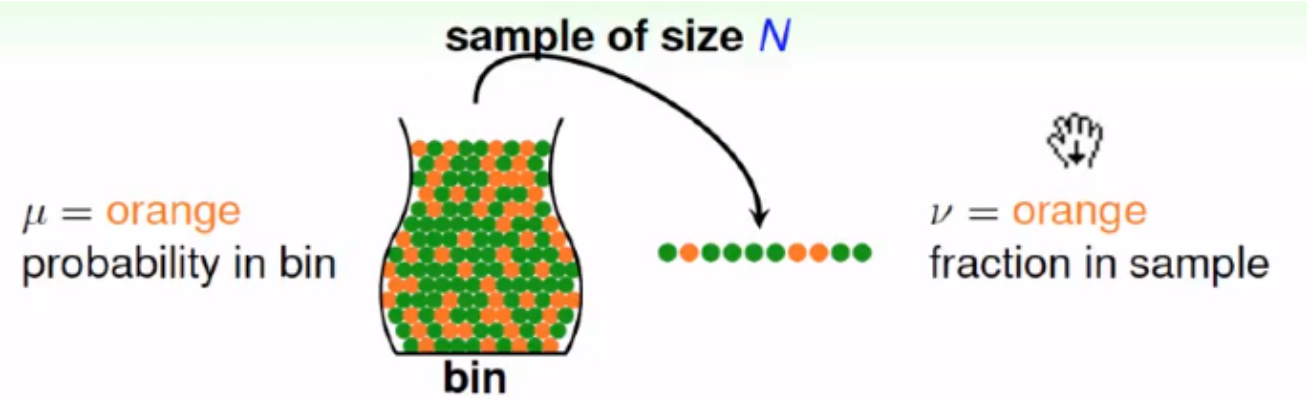
为了介绍VC理论，首先引入Hoeffding 不等式 和 Union Bond（联合界引理）(需要说明的是，为了保证普适性，这两个定理都是非常宽松的，在本文的后面提供的证明中进行说明，这里只给出结论):

Hoeffding 不等式

假设$z_1, z_2, \dots z_N$为N个IID（独立同分布）的随机变量，服从均值为ϕ的伯努利分布:

$\hat{\phi} = \frac{1}{N}\sum_{i=1}^{N} z_i$

则有：$P(|\hat{\phi} - \phi| > \gamma) \leq 2 * exp^{(-2\gamma^2 N))}$,（描述了一组随机变量的均值与期望值相差一个确定值的概率上的upper bound，如下图所示，根据采用得到的v,我们无法得到精确的μ,但Hoeffding 不等式告诉我们，v与μ大于一个确定值ϵ的概率有一个上界，换句话说，只要N足够大，可以保证v与μ一致收敛。



- in big sample (N large), ν is probably close to μ (within ϵ)

$$\mathbb{P}\left[\left|\nu - \mu\right| > \epsilon\right] \leq 2\exp\left(-2\epsilon^2 N\right)$$

需要说明的是：Hoeffding 不等式与中心极限定理不同，对于任意的正整数N都成立。

Union Bond (联合界引理)

假设$A_1, A_2, \dots A_k$表示k个事件（不一定独立）： 则有：

$P(A_1 \cup A_2 \cup .. \cup A_k) \leq P(A_1) + P(A_2) + \dots + P(A_k)$

定义训练集：$D = (x_1, y_1), (x_2, y_2), \dots (x_N, y_N)$,x_i 为来自分布χ的采样,独立同分布,x到y的转化规则为f(x)。

定义固定的假设规则：h(x)

定义规则h的训练误差：$E_{h,in} = \frac{1}{N}\sum_{i=1}^{N} (h(x_i) \neq f(x_i))$

定义规则h泛化误差：$E_{h,out} = \frac{\epsilon}{x \to \chi} h(x) \neq f(x)$ 表示h(x)不等于f(x)的均值。

事件$h(x) \neq f(x)$ 服从伯努利分布，x独立，因此 $h(x) \neq f(x)$ 也独立：从而根据Hoeffding不等式有：

$P(|E_{h,in} - E_{h,out}| > \gamma) \leq 2 * exp^{(-2\gamma^2 N))}$

因此对于一个固定h(x)，当N足够大时，即使不知道f，也可以从$E_{h,in}$ 推断$E_{h,out}$，从而保证训练误差与泛化误差足够接近,即当$E_{h,in}$足够小时，只要N足够大，则$E_{h,out}$就会很小（即在我们未看过的数据上也可以有很好的预测），从而可以认为机器学到了一些规则。

但是如果只有一个固定的h(x)，学习算法无法保证$E_{h,in}$足够小。我们假设假设空间有M个不同的规则h{1...M},定义事件A_j :$|E_{h_j,in} - E_{h_j,out}| > \gamma$则：$P(A_j) \leq 2 * exp^{-2\gamma^2 N}$ ，学习算法利用给定的误差衡量方式选取h，我们无法保证会选到哪个h，因此为了保证机器能学到规则，需要保证对于假设空间H中任意的h_j ，都有$|E_{h_j,in} - E_{h_j,out}| > \gamma$ ，事件A_j表示对于规则h_j $|E_{h_j,in} - E_{h_j,out}| > \gamma$ ，因此，需要保证对于假设空间H中任意的h_j，都有$|E_{h_j,in} - E_{h_j,out}| > \gamma$的概率等价于$P(A_1 \cup A_2 \cup .. \cup A_k)$

则根据联合界引理可以得到：$P(A_1 \cup A_2 \cup .. \cup A_k) \leq \sum_{i=1}^{k} P(A_j) \leq \sum_{i=1}^{k} 2 * exp^{-2\gamma^2 N} = 2 * k * exp^{-2\gamma^2 N}$

通过上式，可以得到，当k为有限个时，只要N足够大，我们依然可以保证$E_{H,in}$与$E_{H,out}$ 足够接近，即保证机器可以学到一些规则。即：

只要$N \geq \frac{1}{2*\gamma^2} log^{\frac{2k}{\delta}}$ (其中$\delta = 2k * exp^{-\gamma^2 N}$) 就可以保证：$P(|E_{H,in} - E_{H,out}| \leq \gamma) \geq 1 - \delta$,即$E_{H,in}$与$E_{H,out}$ 实现一致收敛。

同时，我们根据上式也可以定义样本复杂度和误差界限

样本复杂度

给定γ和 δ,求解 N（即在某个置信度内要达到某个特定的误差界限，需要多少训练样本）：

根据上式可以得到 ：$N \geq \frac{1}{2*\gamma^2} log(\frac{2k}{\delta})$ ，即只要$N \geq \frac{1}{2*\gamma^2} log(\frac{2k}{\delta})$，那么在$1 - \delta$的概率界内（置信度），可以保证对于任意的h属于H，$|E_{h,in} - E_{h,out}| > \gamma$

误差界限

给定N和δ,求解γ（对于固定样本N,在一定的置信度内，可以保证训练误差和泛化误差在一个界限内），$|E_{h,in} - E_{h,out}| \leq (\frac{1}{2N} log^{\frac{2k}{\delta}})^{\frac{1}{2}}$

有限假设空间偏差方差权衡

定义 $\hat{h} = argmin(E_{h,in})$,($\hat{h}$为假设空间中使训练误差最小的假设规则)

定义 $\tilde{h} = argmin(E_{h,out})$ ($\tilde{h}$为假设空间H中使泛化误差最小的假设规则),$E_{\tilde{h},out}$ 描述了假设空间误差的偏差

则有$E_{\hat{h},out} \leq E_{\hat{h},in} + \gamma \leq E_{\tilde{h},in} + \gamma \leq E_{\tilde{h},out} + \gamma + \gamma \leq E_{\tilde{h},out} + 2\gamma$

第一个和第三个小于等于号的证明用到了$|E_{h,in} - E_{h,out}| \leq \gamma$ (对任意h都成立)

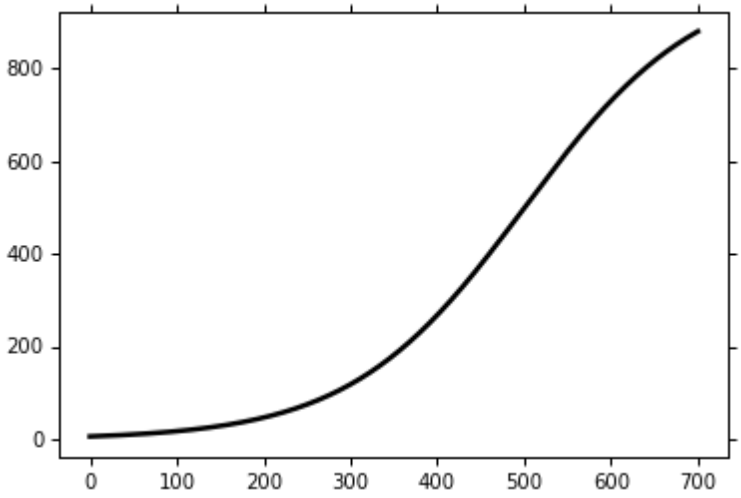
第二个小于等于号用到了$\hat{h}$ 的定义$\hat{h} = argmin(E_{h,in})$

定理：假设假设空间的个数为k,样本数量N,置信度δ固定，那么：

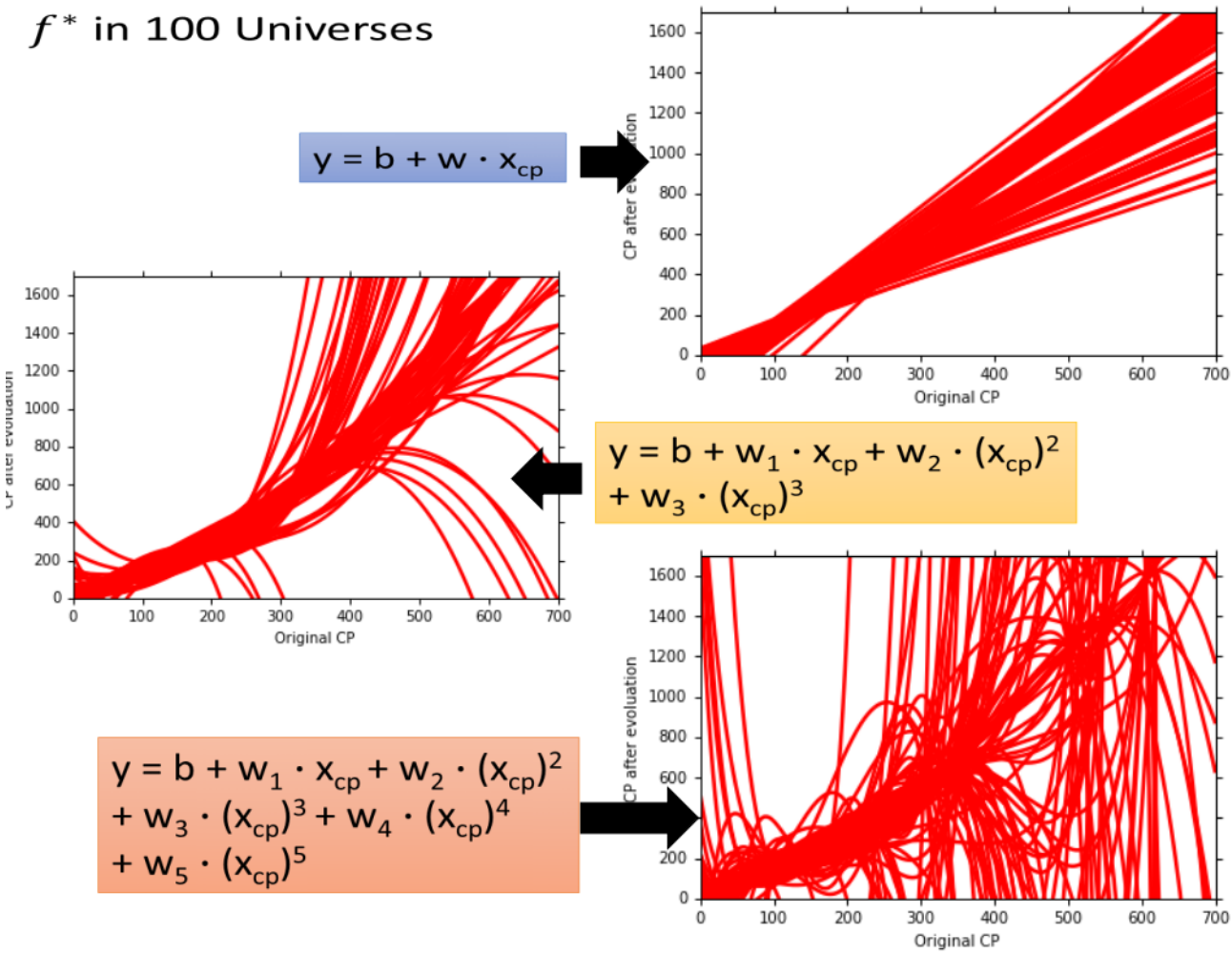
$$E_{\hat{h},out} \leq E_{\hat{h},in} + \gamma \leq E_{\tilde{h},in} + \gamma \leq E_{\tilde{h},out} + \gamma + \gamma \leq E_{\tilde{h},out} + 2(\frac{1}{2N} log^{\frac{2k}{\delta}})^{\frac{1}{2}}$$

上式中最后一项中的第一项为假设空间误差的偏差（假设空间误差的最优误差与贝叶斯最优误差的差），第二项描述了假设空间误差的方差(假设空间误差的分散程度)，k越大，即假设空间越大，那么假设空间所能达到的最优误差可能越小，距离贝叶斯最优误差的差可能会越小，但是k增大也会增大假设空间误差的方差，k较小时，可以保证假设空间的方差很小，但不能保证假设空间的偏差足够小（假设空间的最优误差与贝叶斯最优误差之间的差距），这就是机器学习中的偏差-方差权衡问题。

为了更直观的说明，偏差与方差的问题，以多项式回归为例进行分析，假设下图为要模拟的模型，在该模型基础上添加部分随机噪声，然后进行采样，获得多组训练集。

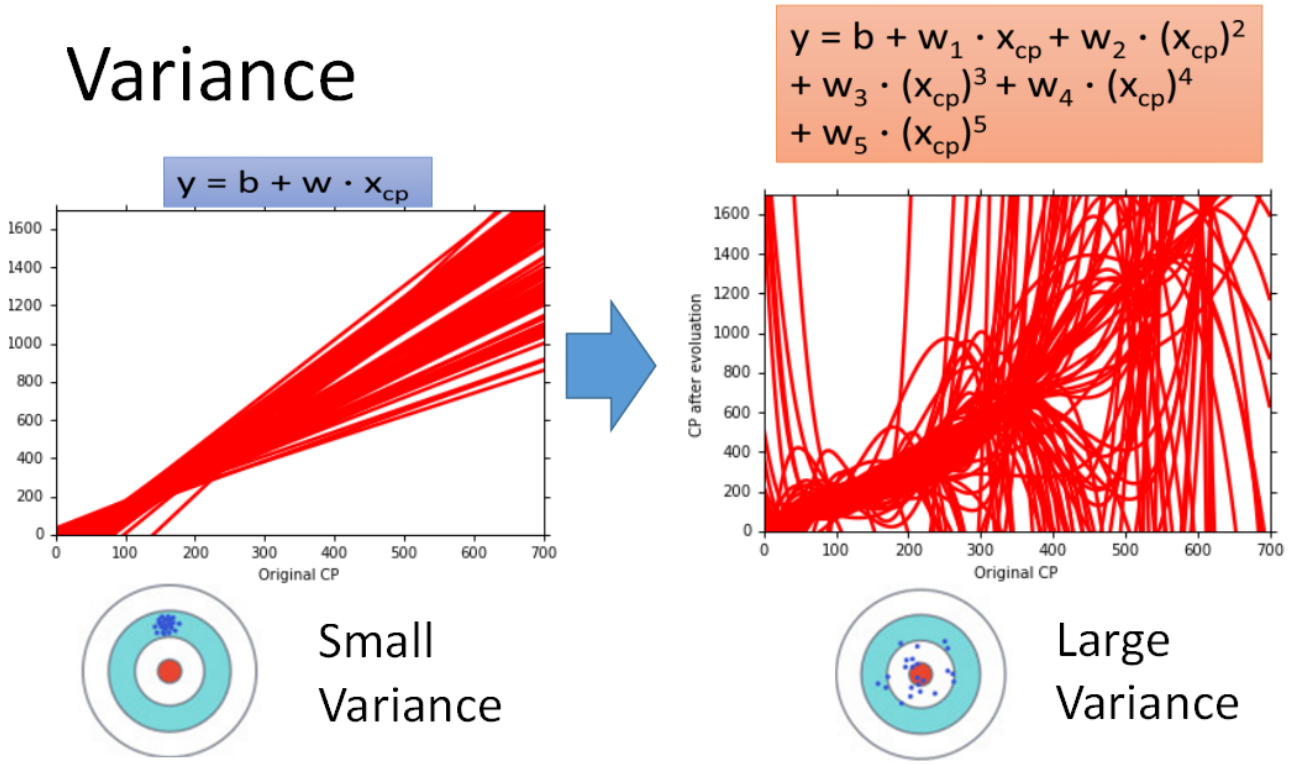


利用上述方法生成的多组训练集，采用不同多项式的假设空间进行学习。如下图所示，分别为采用一次方，三次方，五次方假设空间进行学习，得到如下结果（每一条线表示利用不同的训练集得到的使训练误差最小的模型）。

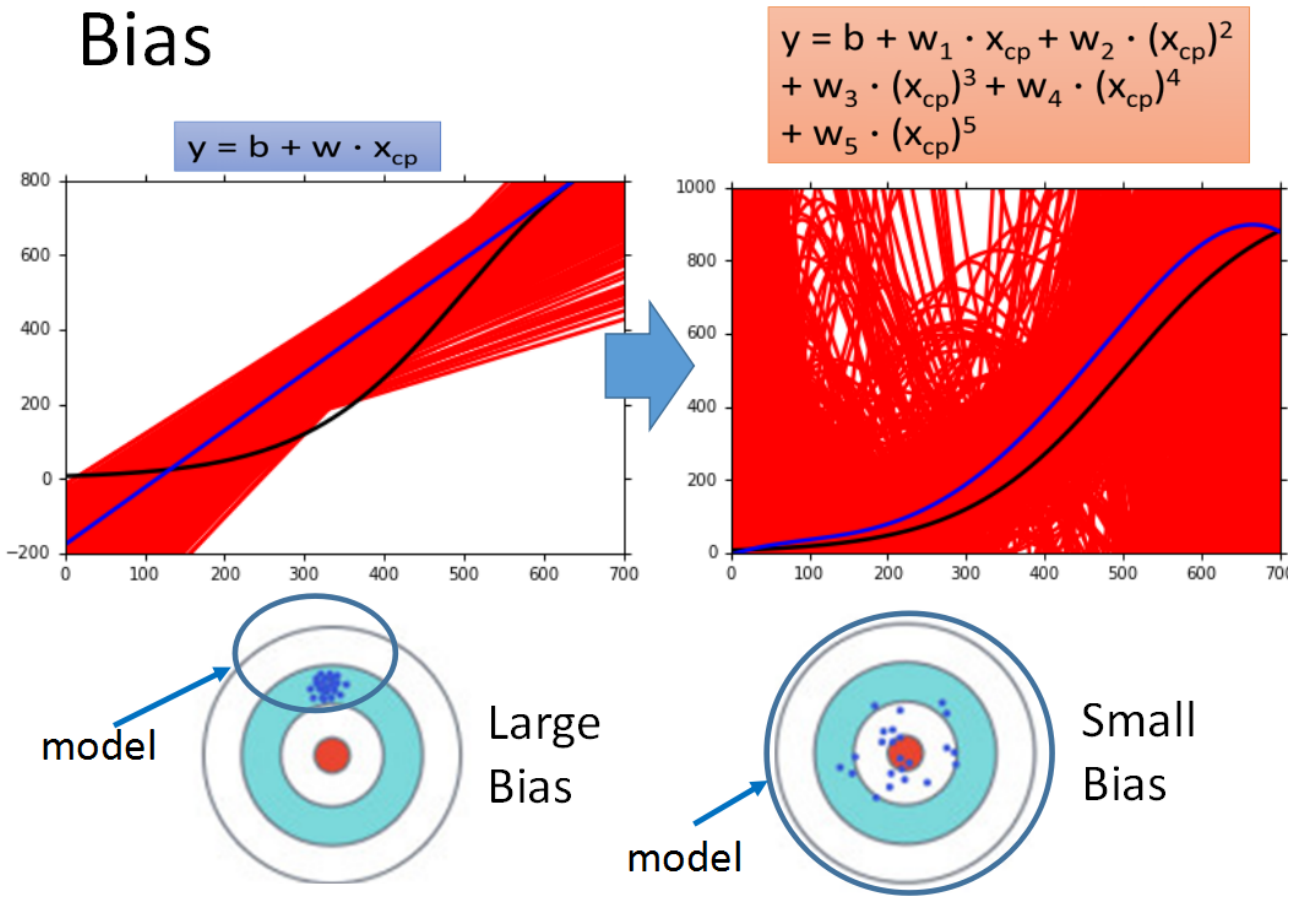


从上图可以看到，模型的自由变量越少，即模型的d_{vc}越小，利用不同的训练集得到的模型越集中，d_{vc}越大，不同训练集训练得到的模型差别越大。

将模型的参数进行可视化（这里只是示意图），得到如下结果。



假设图中红色中心为最优模型，左图d_{vc}较小，不同训练集得到的模型相对集中，方差小，右图d_{vc}较大，不同训练集得到的模型相差比较大，比较分散，对应于高方差。



上图中，黑色线为理想的模型，蓝色线为不同训练集模型的平均，可以看到，左图蓝色线与理想模型相差较远，对应于高偏差，右图中蓝色线与理想模型相差较小，对于与低偏差。

总结上述试验：

当模型d_{vc}太小时，训练误差与泛化误差接近，但训练误差本身就比较大，对应于underfitting，泛化误差主要来自偏差。

当模型d_{vc}太大时，训练误差较小，但泛化误差比较大，对于对overfitting，泛化误差主要来自方差。

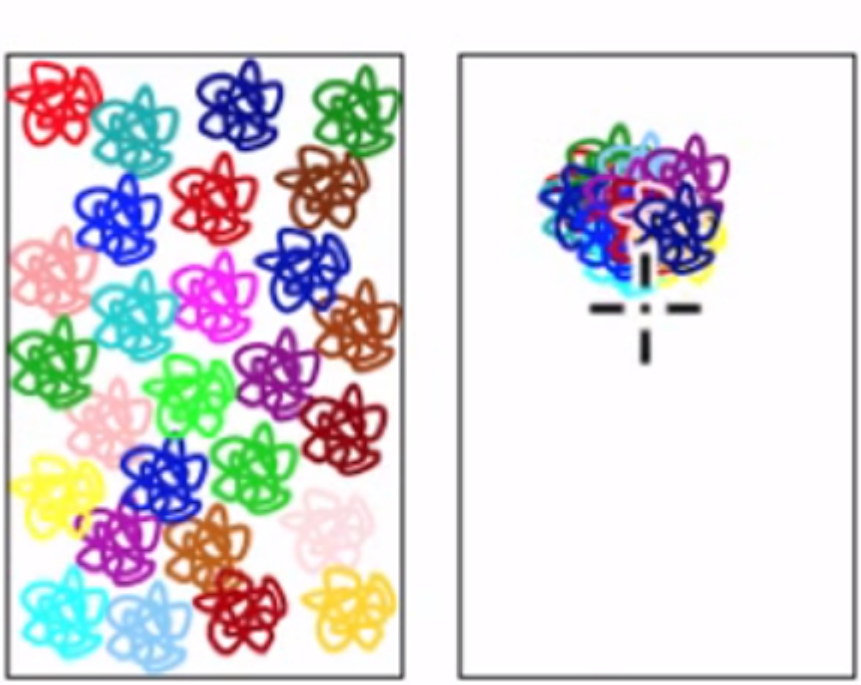
无限假设空间的学习保证

上一节推导了有限假设空间内的学习保证，我们得到，

$$P(|E_{H,in} - E_{H,out}| \geq \gamma) \leq 2 * k * exp^{-2\gamma^2 N}$$

如果k无限，即使N再大，也不能保证$E_{H,in}$与$E_{H,out}$的一致收敛，事实上，对于大部分学习算法，只要参数动一点，那么可以认为是不同的规则，因此大多数算法的假设空间有无限多个，因此，按照上述定理，并不能保证一致收敛，但是上式推导中，使用到了联合界引理，这是一个非常宽松的上界。

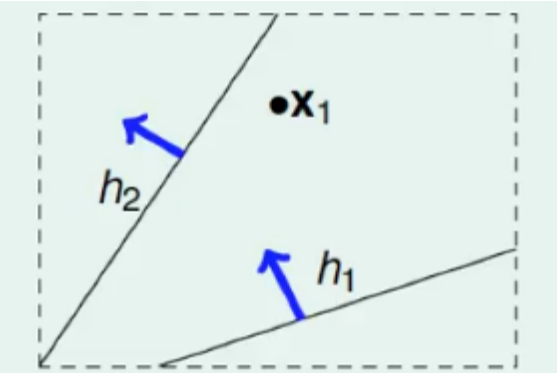
$P(A_i \cup A_j) = P(A_i) + P(A_j) - P(A_i \cap A_j)$ ，只有在$P(A_i \cap A_j) = 0$时，上式才成立。事实上对于假设空间中相似的规则（函数），$P(A_i \cap A_j)$ 可能远大于0，因此，如下图所示，$P(A_1 \cup A_2 \cup .. \cup A_k)$可能远小于 $P(A_1) + P(A_2) + ... + P(A_k)$



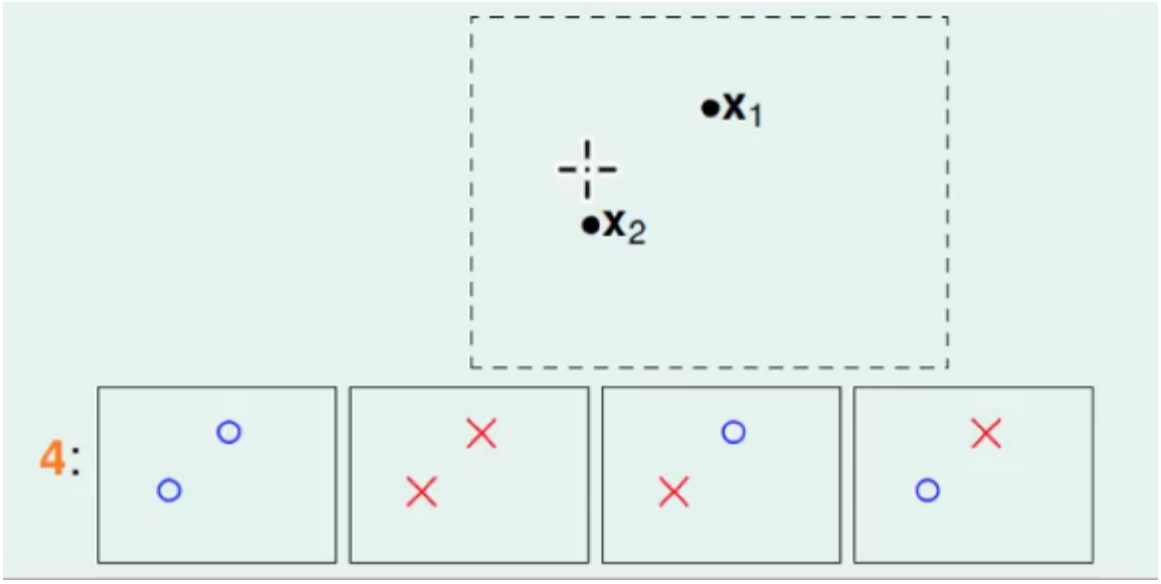
因此，我们需要找一个等效的描述假设空间复杂度的参数，对于二分类来说这就是VC维，接下来，我们将简单阐述VC维，并说明为什么VC维可以代表假设空间的复杂度。

为了寻找等效的描述假设空间复杂度的参数，我们不用具体的假设空间函数的个数，而采用假设空间的种类。

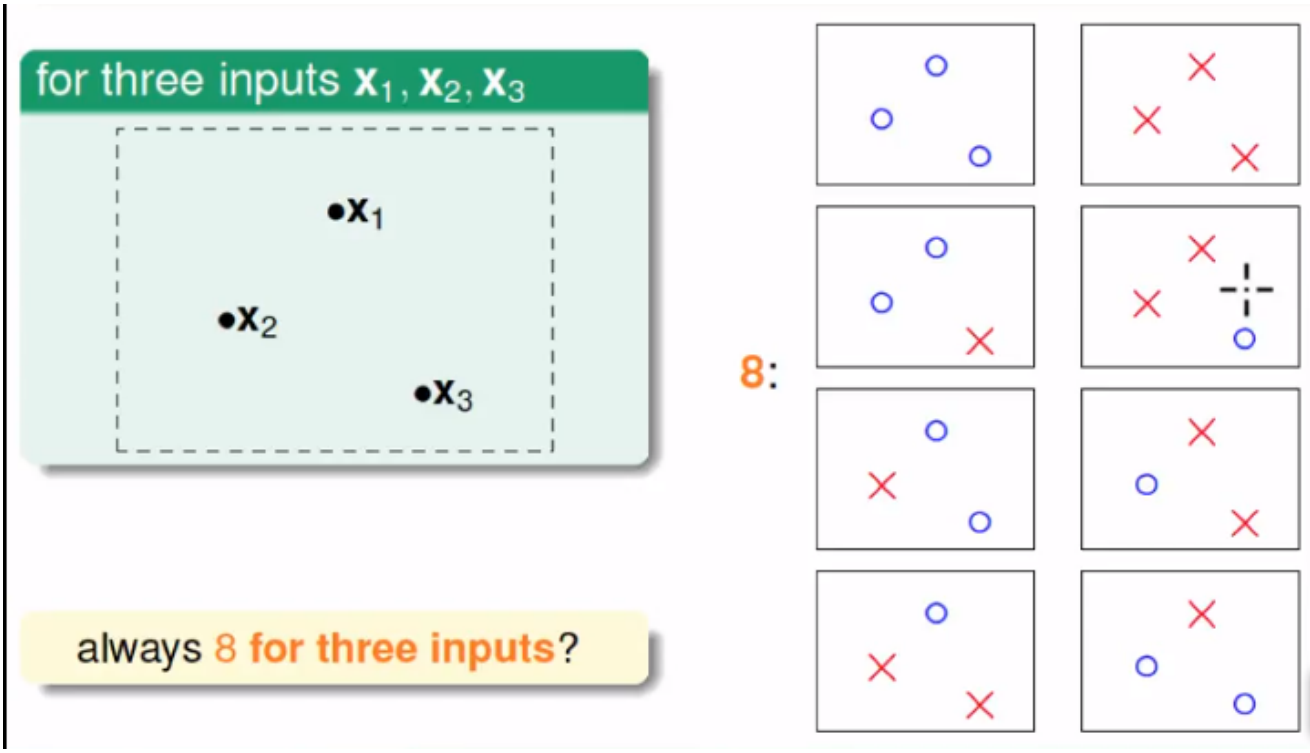
为此我们定义dichotomia：表示假设空间对有限的N个点进行不同标记的种类数（对于二分类问题，最多有2^N种可能,N个点，每个点都两种可能）如下图所示，对于二维的perceptron模型，如果N为1，则有两种可能性（0,1），N为2，则有4种可能性，((1,1),(1，0),(0,1),(0,0)),如果N为3，则会有8中可能性，如果N为4，则会有14中可能性（分数圈中的标记方式无法使用二维perceptron模型进行分割，对应的还有7种情况），此时不为2^4。


$\bullet x_1$
h_2
h_1

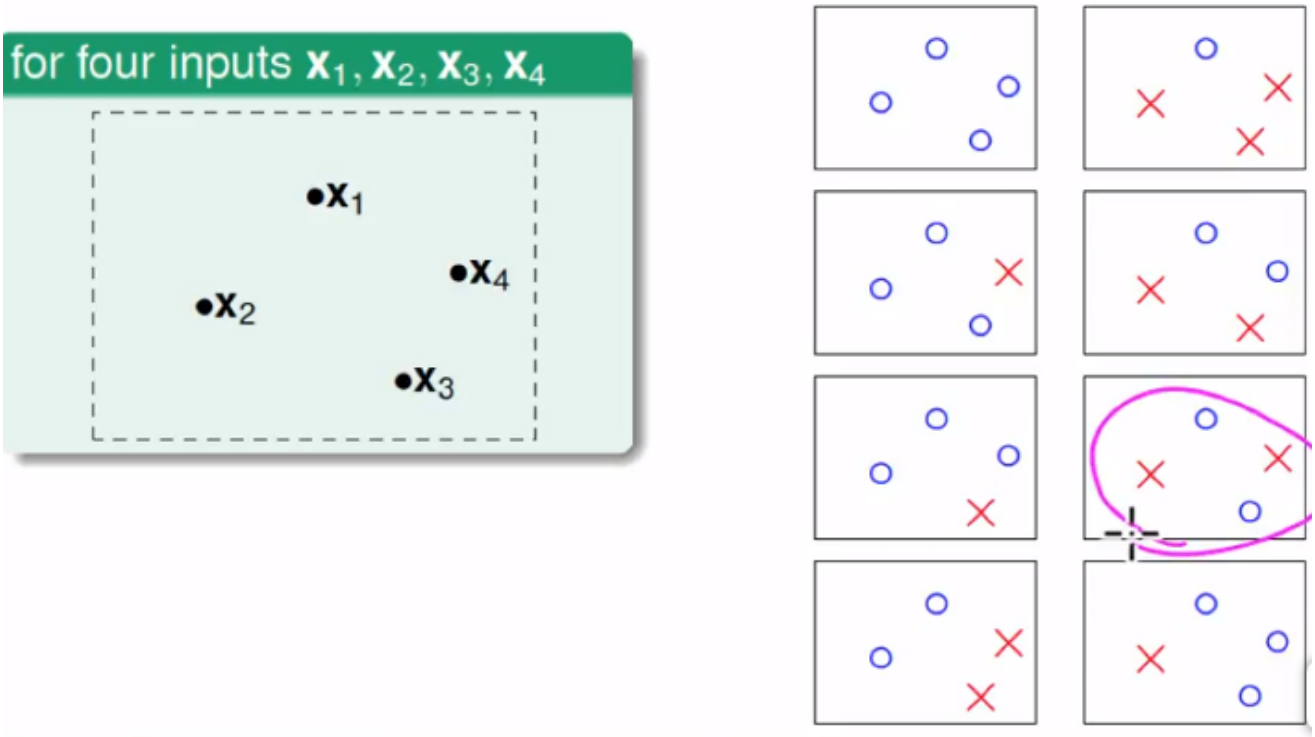
N=1


$\bullet x_1$
$\bullet x_2$
4:

N=2


for three inputs x_1, x_2, x_3
$\bullet x_1$
$\bullet x_2$
$\bullet x_3$
always 8 for three inputs?
8:

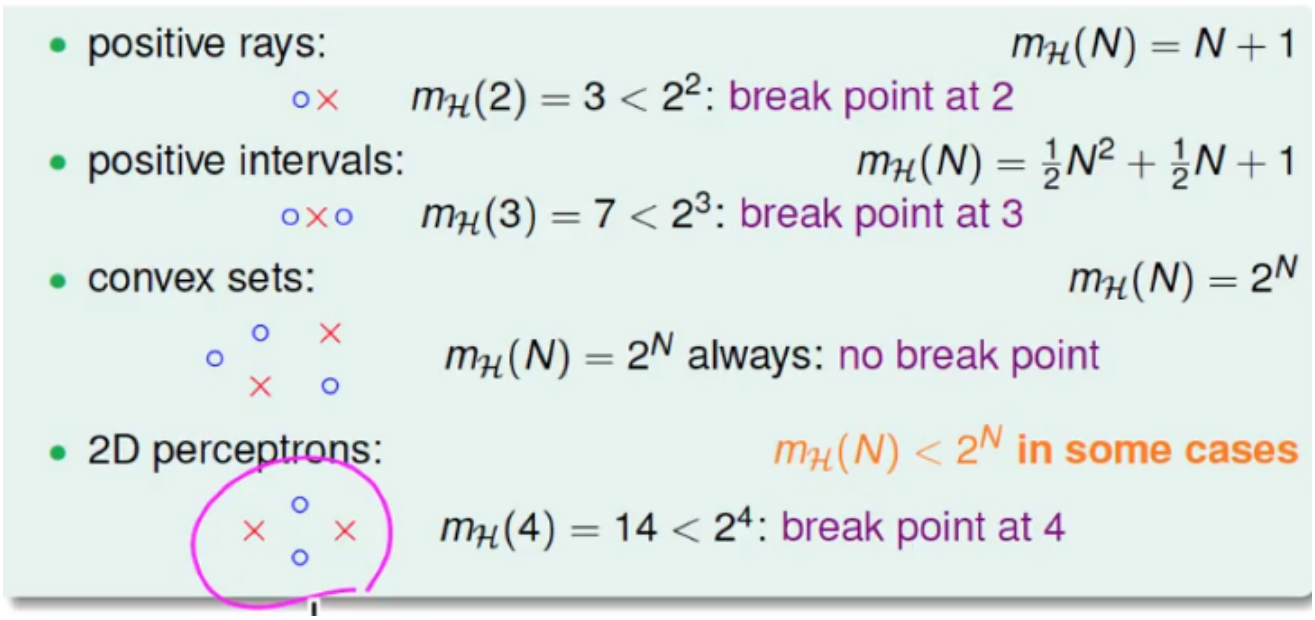
N=3



N=4

接下来，我们定义成长函数m(N):假设空间H对空间内N个点，进行标记的最大的种类数（即最大的dichotomia，例如，对于二维空间中4个点，如果四个点连成一条线，此时dichotomia为5，如果任意三个点都不在一条直线上，dichotomia为14，m(N)就是这个最大值14），因此成长函数是dichotomia的上界。

接下来，借用成长函数定义break point点 k：最小的使得m(k) < 2^k的k值，例如，对于二维的perceptron模型，N=3时，可以做出全部的$2^3 = 8$种可能，但是如果N=4,就无法做出全部的$2^4 = 16$种可能。因此这里break point就等于4.下面是不同类型简单模型的break point数目。



可以证明，如果假设空间没有break point点，那么m(N) = 2^N ,如果存在break point点，那么可以证明$m(N) = O(N^{k-1})$,关于这个的证明，这里不做论述（事实上我也没有证明），这里的k-1即为该假设空间的VC维。

下面给出VC维的定义：d_{vc}表示能够被shatter 的最大集合个数，这里的shatter表示H能对N个点进行2^N 种的不同标记。

当然假设空间的种类不能直接作为k的替换值，还需要经过很多推导(但我认为最重要的是理解VC维的意义，以及为什么VC维可以作为模型复杂度的度量）。

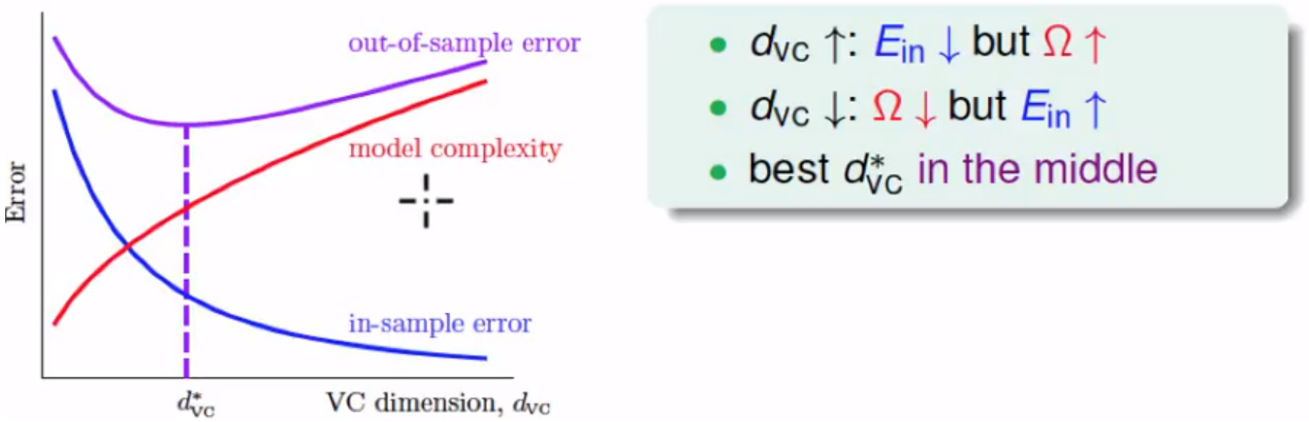
经过推导，可以得到，对于H中任意的h：$P(|E_{h,in} - E_{h,out}| \geq \gamma) \leq 4 * (2N)^{d_{vc}} * exp^{-\frac{1}{8}\gamma^2 N}$

从式中可以看到，第一项$(2N)^{d_{vc}}$随着N的增长以多项式增长，而后一项$exp^{-\frac{1}{8}\gamma^2 N}$随着N以指数方式下降，因此，当N足够大时，可以保证$E_{\hat{h},in}$与$E_{\hat{h},out}$ 实现一致收敛。

定义$\delta = 4 * (2N)^{d_{vc}} * exp^{(-\frac{1}{8}\gamma^2 N)}$

则有：$E_{h,out} \leq E_{h,in} + sqrt(\frac{8}{N} log^{\frac{4(2N)^{d_{vc}}}{\delta}})$

大多数情况下，d_{vc} 等价于模型的自由度（即模型拥有的参数），对于线性分类模型，n维模型，包含n个w和1个b，自由度和d_{vc}都为n+1。



上图刻画了随着d_{vc}的增加，$E_{\hat{h},in}$不断减小（$E_{\hat{h},in}$可以近似$E_{\tilde{h},out}$（即模型所能达到的最优的泛化误差），因此，可以作为偏差项的估计），即随着模型d_{vc}的增加，偏差越来越小。但是随着d_{vc}的增加，方差项越来越大（第二项），因此，最佳的模型在中等程度的VC维处取得。d_{vc}之前的区域称为under fitting，d_{vc}太小，无法使$E_{\hat{h},in}$足够小，对应高偏差，右边称为over fitting，d_{vc}太大，$E_{\hat{h},in}$ 很小，但是$E_{\hat{h},out}$太大，不能很好的generalization（不能在未看过的数据上做出很好的预测），对应高方差。

需要说明的是，上式是一个非常宽松的上界，不应该拿来作为对于需要数据量的估计。

通过上面的论述，我们已经对d_{vc}的定义，以及为什么可以作为模型复杂度的度量有了清晰的认识，对于偏差和方差的权衡也有比较好的理解，接下来，我将会简单阐述，为什么正则化可以有效减弱overfitting。

从VC 维的角度理解正则化

从第一节机器学习框架，可以得知如果没有正则化，机器学习就是通过最小化定义的error 函数，(对于二分类问题来说，就是在假设空间中选择能最小化E_{in}，或者E_{in}的近似误差的规则（函数））。而正则化是对最小化的目标函数添加某些约束，这些约束无法从数据中获得，代表了先验的偏好，例如，L2 的正则化，可以表述为：$min E_{in}, s.t \sum w_i^2 \leq C$,可以理解为L2 正则化 只在w值较小的区域内选取），因此有效的d_{vc}变小，假设空间误差的方差也变小，减小过拟合的风险。

总结来说正则化提供了调节假设空间d_{vc}的有效手段。

另一种理解如下， 对于约束最优化问题，$minE_{in}, s.t \sum w_i^2 \leq C$ 可以通过拉格朗日乘数，转换为无约束的最优化问题，即：$E_{in} + \frac{\lambda}{2} w^T w$,将上式命名为E_{aug},因此，如果λ 选取的合适，E_{aug}可以认为与E_{out}更接近，因此通过最小化E_{aug}获得的h，可以使E_{out}更小，即有更好的泛化能力。

深度学习与VC维

VC维是机器学习领域一个非常基础的概念，为许多机器学习方法的可学习性提供了坚实的理论基础。深度学习的VC维可以表示为:$d_{vc} = O(VD)$,V为神经元个数，D为weight的个数。但是上式是针对全连接网络而言的，事实上CNN通过稀疏连接和权值共享机制（和各种精巧的连接机制），使得模型的参数量并不是很大，事实上模型的自由度只与卷积核大小和个数有关；RNN中也存在参数共享机制，使得其参数量也不是很大，但深度神经网络参数耦合在一起，使得深度学习在参数量并不是很大的情况下，也可以有非常强大的表达能力。

事实上许多时候深度学习的参数量要远比常规机器学习参数量更少，比如语言模型中，n-gram模型和矩阵分解的方法需要的参数量，要远大于RNN参数量，而且使用gated RNN，如LSTM,GRU等，可以捕获更长的时间序列信息，因此，**深度学习其实更不容易overfitting** 。

总结来说，深度神经网络的**高度可扩展性** 和通过**将优化和特征提取工作绑定在一起** 的方式提取特征，使得提取的特征更加有效，可以获得更低的训练误差。同时由于深度神经网络中的**参数耦合**，使得深度学习在参数量不是很大的情况下（比较低的d_{vc}），也可以有非常强大的表达能力，使得泛化误差与训练误差足够接近，从而保证深度学习获得更好的泛化能力，实践上的巨大成功也证明了这点。

深度学习中的正则化与VC维解释

在深度学习中应用正则化有以下几种方式：

1、L2正则化

2、L1正则化

3、early stopping

4、稀疏连接与参数共享

5、使用max-margin loss

6、dropout

7、batch norm

L2 正则化

L2正则化表达了，我们希望权重尽可能小的先验知识

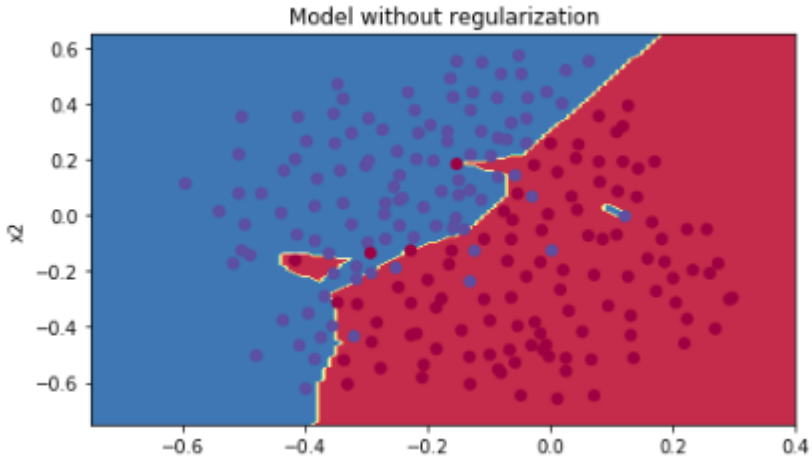
$$L_{reg} = L_{org} + \frac{\lambda}{2N} \sum_l \sum_k \sum_j (W_{kj}^l)^2$$

其中 L_{org}为不做规则化的cost函数，L_{reg} 为正则化后的cost函数，W_{kj}^l为连接第l层第k个神经元与第l-1层第j个神经元的权重，（需要说明的是，由于W一般是很高维的向量，偏置b相对W可忽略，因此，一般不加入到规则化的cost函数中）

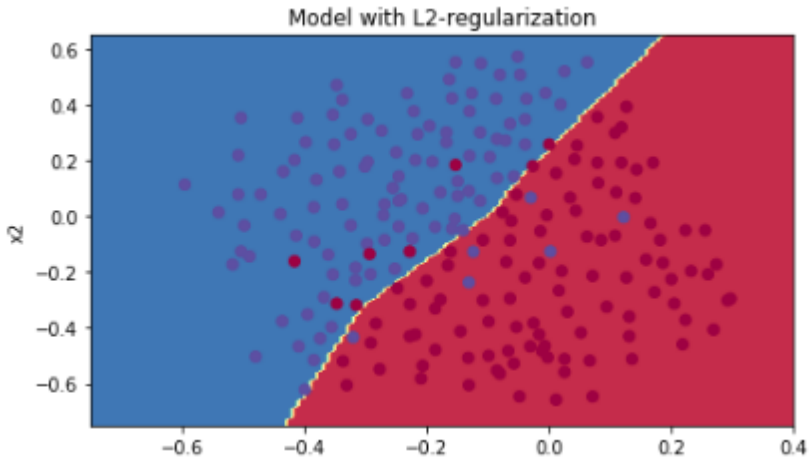
经过推导，可得到更新过程如下：

$$W^{t+1} = (1 - \eta\lambda)W^t - \eta \frac{\alpha L_{reg}}{\alpha W^t}$$

从上式可以看到：更新W时，除了梯度项，还会还会减去一个$\eta\lambda W^t$,如果W^t比较大，那么会减去一个比较大的值，如果W^t比较小，则会减去一个比较小的值，因此经过多轮更新后，W平均都比较小，但0值也很小。



不做正则化分类边界



L2正则化后分类边界

上边两张图分别为不做正则化和L2正则化后的分类边界，可以明显看到，L2正则化后边界更加平滑，而原始分类边界则存在许多异常点，存在过拟合的风险。

不做正则化训练集准确率可以达到94.8 %，但测试集准确率只有91.5%，存在明显的过拟合问题。

L2正则化后训练集准确率为93.8%，测试集的准确率为93%，基本不存在过拟合，测试集准确率明显提高。

（该例子来自吴恩达深度学习课程作业）

L1 正则化

L1正则化表达了我们希望连接尽可能稀疏的先验知识

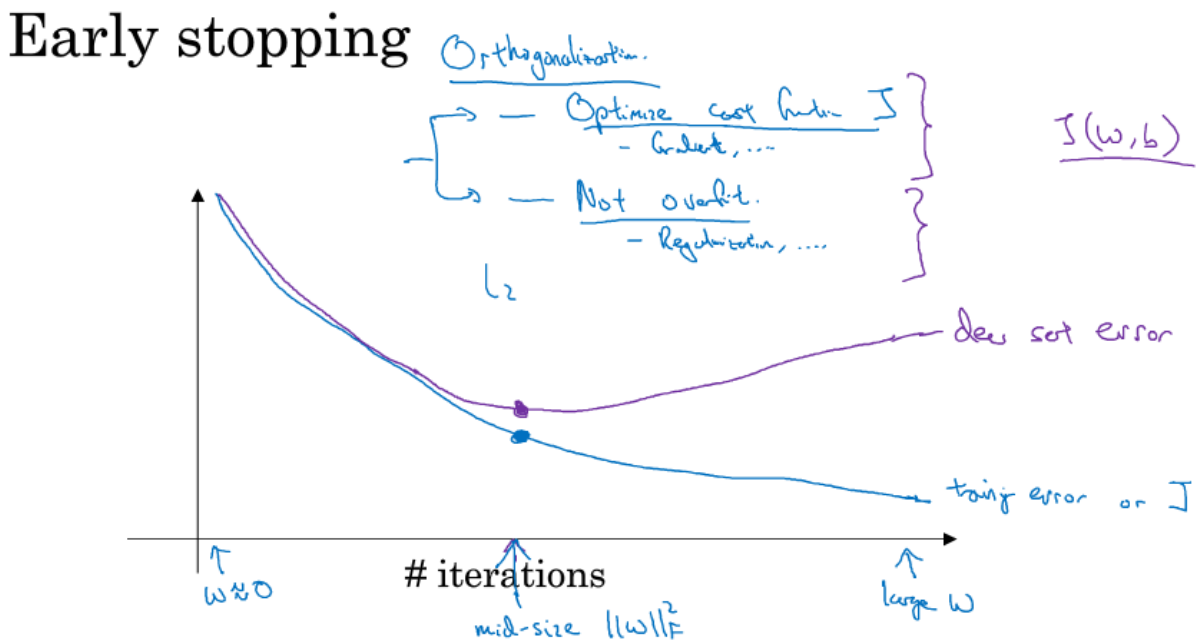
$$L_{reg} = L_{org} + \frac{\lambda}{2N}\sum_l\sum_k\sum_j|W_{kj}^l|$$

经过推导，可得到更新过程如下：

$$W^{t+1} = W^t - \eta\frac{\alpha L_{reg}}{\alpha W^t} - \eta\lambda sign(W^t)$$

上式中sign表示取正负号，从上式可以看到，更新W时，除了梯度项，如果W为正，那么会固定减去$\eta\lambda$,如果W为负，则会加上一个$\eta\lambda$,因此，经过多轮更新后，会有大量W变为0，即产生稀疏解。

early stopping



Andrew Ng

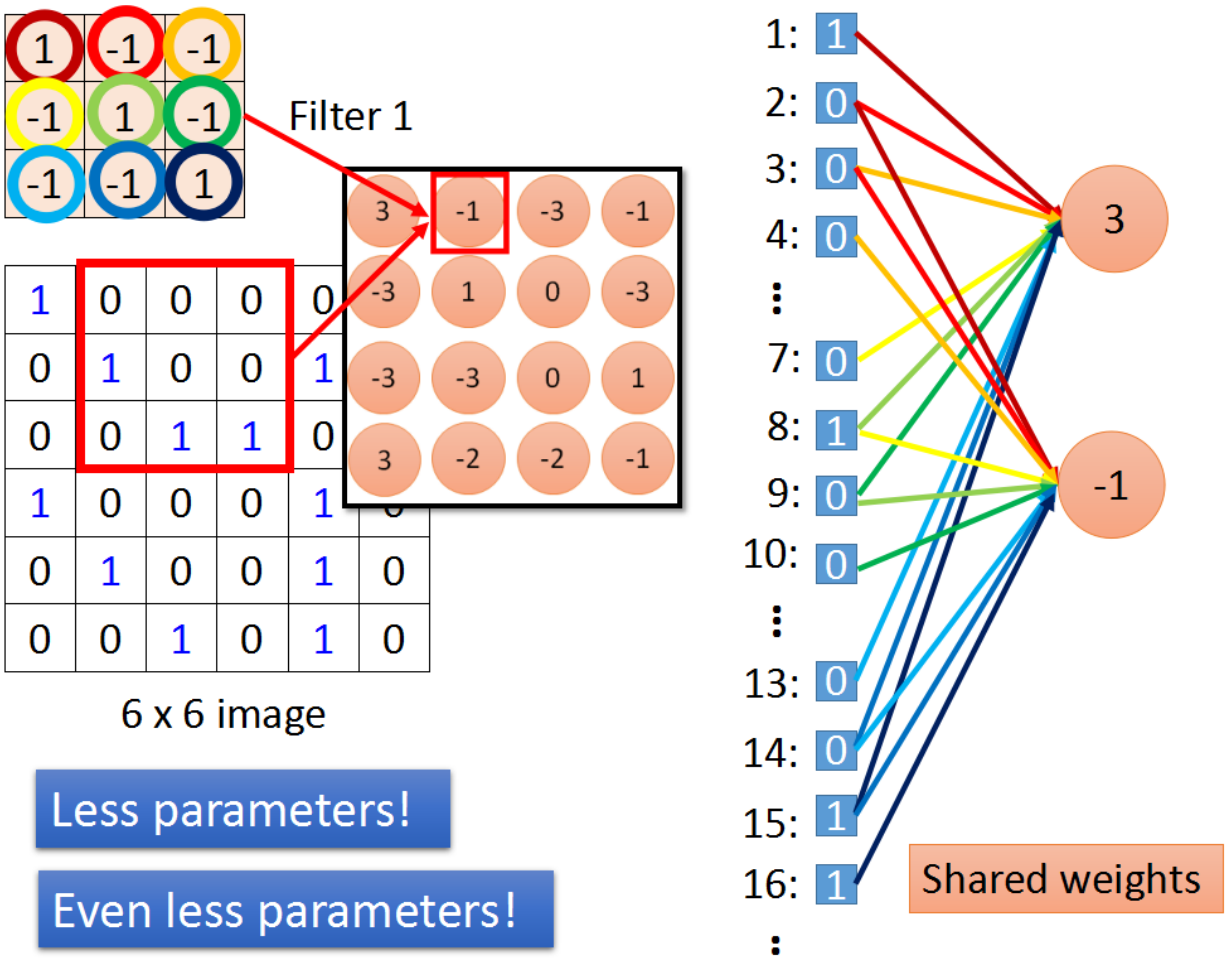
如上图所示，随着训练程度的增加，训练集上的误差不断减小，但验证集上的误差确呈现先下降后上升的缺失，因此，为了获得最好的泛化误差，可以进行early stopping。

接下来尝试说明，为什么early stopping可以有效减弱overfitting：

early stopping 是通过控制训练步数来控制模型的有效容量，可以理解为将优化的参数空间的初始值限定在一个小的领域内，因此可以理解为一种特殊的L2正则化，但early stopping 能检查训练集合验证集的测试误差，自动确定正则化的大小，而L2正则化则需要进行多组超参数测试，因此更具优势。

参数绑定与参数共享

L2正则化表达了，我们希望权重尽可能小的先验知识，L1正则化表达了我们希望连接尽可能稀疏的先验知识，有时我们可能无法精确地知道应该使用什么样的参数，但根据相关领域和模型结构方面的知识得知模型之间应该存在一些相关性，比如迫使某些参数相等，这种方式称为参数共享，例如卷积神经网络中，图像数据中由于某些有意义的特征只需要局部信息，同样特征可能存在不同的区域等特性采用局部连接和参数共享机制。事实上卷积神经网络可以理解为全连接网络施加了非常强的正则化。



max-margin loss

与SVM中精神相同，深度学习中，将loss函数定义为最大误差,然后通过优化方法**最小化最大误差** ，只选择能使最大误差最小的模型。

例如在window classification: 在邻近单词的上下文窗口中对单词进行分类：

loss函数，$J = max(0, 1 - s + s_c)$ ：

使得真实窗口计算出分数更高，不是真实窗口计算出来的分数更低，而且是真实窗口的分数比计算出来的分数足够好，大于1。边界更加稳健，获得更好的泛化能力。

集成提升泛化能力的数学基础

上文中提到，正则化通过调节假设空间d_{vc}来实现提升泛化能力，事实上dropout与batch norm提升泛化能力的机制与其他方法并不完全一致，可以将dropout，BN等看作一种集成的机制，泛化能力的提升主要来自多样性的增加，下面对集成方法提升泛化能力的数学基础做出说明（下述证明来自周志华<机器学习>第8章集成学习，对其中的证明添加了中间几步，使证明更加清晰）。

假定我们有k个学习器：$\{h_1, h_2, \dots h_k)\}$,通过加权平均结合缠身的集成来完成回归任务f(其他任务类似，这里重点说明集成的精神)。集成结构为H,对事例x，

定义学习器h_i的分歧(ambiguity)为：$A(h_i|x) = (h_i(x) - H(x))^2$

则集成的分歧为$\tilde{A}(h|x) = \sum_{i=1}^{k} w_i A(h_i|x) = \sum_{i=1}^{k} w_i((h_i|x) - H(x))^2$

这里的分歧表示个体学习器在样本x上的不一致性，反映了个体学习器的多样性。

个体学习器h_i的平方误差为：$E(h_i|x)=(f(x)-h_i(x))^2$

集成H的平方误差为：$E(H|x)=(f(x)-H(x))^2$

个体学习误差的加权平均值：$\tilde{E}(h|x)=\sum_{i=1}^{k}w_i.E(h_i,x)$

将$\tilde{A}(h|x)$配方,则有$\tilde{A}(h|x)=\sum_{i=1}^{k}w_i(h_x(x)-H(x))^2=\sum_{i=1}^{k}w_i(h_i(x)-f(x)+f(x)-H(x))^2$

$\tilde{A}(h|x)=\sum_{i=1}^{k}w_i(h_i(x)-f(x))^2-2*w_i*(f(x)-h_i(x))*(f(x)-H(x))+(f(x)-H(x))^2$

$H(x)=\sum_{i=1}^{k}w_ih_i(x)$

从而有

$\tilde{A}(h|x)=\sum_{i=1}^{k}w_i(h_i(x)-f(x))^2-2(f(x)-H(x))^2+(f(x)-H(x))^2=\sum_{i=1}^{k}w_i(h_i(x)-f(x))^2-(f(x)-H(x))^2$

$\tilde{A}(h|x)=\sum_{i=1}^{k}w_i(E(h_i|x)-E(H,x))=\tilde{E}(h|x)-E(H|x)$

对全体样本则有(根据均值定义)：

$\sum_{i=1}^{k}w_i\int A(h_i|x)p(x)dx=\sum_{i=1}^{k}w_i\int E(h_i|x)p(x)dx-\int E(H|x)p(x)dx$

令$\tilde{E}=\sum_{i=1}^{k}w_i\int E(h_i|x)p(x)dx$,个体学习器的泛化误差加权均值

令$\tilde{A}=\sum_{i=1}^{k}w_i\int A(h_i|x)p(x)dx$个体学习器的加权分歧值(所有数据上)

从而有$E(H|x)=\tilde{E}-\tilde{A}$

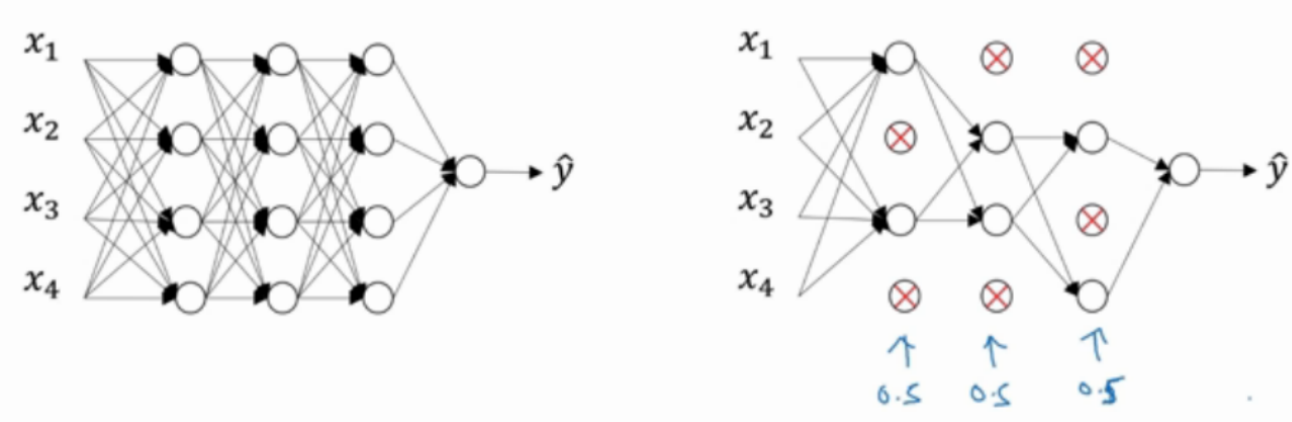
上式清楚的说明了，只要个体学习器泛化误差越小，个体学习器之间的分歧越大，则集成模型的泛化误差就越小。

上式清晰的表明，增加模型的多样性，可以减小集成模型的泛化误差。

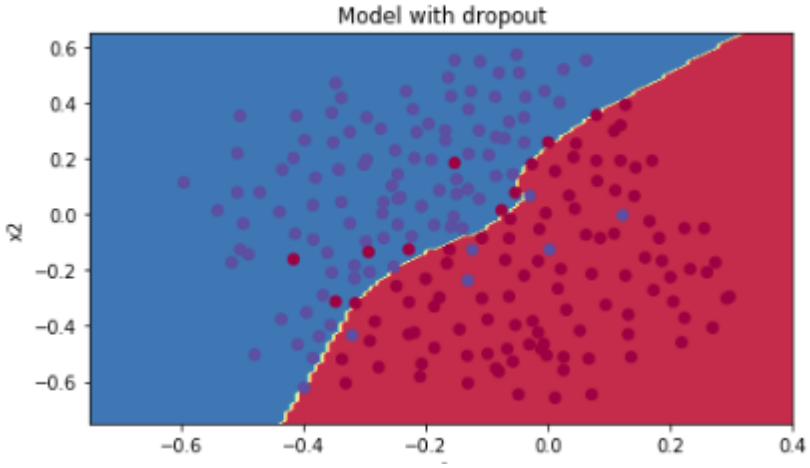
Dropout

L2,L1正则化通过降低假设空间d_{vc}，从而避免过拟合，Dropout能够有效减弱过拟合的风险是因为它相当于集成了很多个不同的神经网络，上面说明了集成提升泛化能力的数学基础，下面介绍dropout。

Dropout原理



如上图所示，Dropout 可以施加在任何隐藏层，假设对某一隐藏层施加keep_prob的dropout，则每一次迭代过程中，使该层中的每个神经元以(1 - keep_prob)的比例失活（ shut down ）（ 失活的神经元不参与本次参数迭代的forward和backward ）。



dropout的分类边界

相比于不做dropout的分类边界，相对比较平滑，测试集上的准确率达到了95%，相比91%有明显的提高。

对某隐藏层施加dropout后，那么被施加dropout的每一层的每个节点被有可能失活，因此相当于训练了多个网络（假设神经元总数为N，则会有2^N种可能网络（输出不会过分依赖任何一个神经元的输出，减少了数据扰动产生的影响））。因此，dropout可以被认为是集成大量深层神经网络的bagging方法（当然，与完全训练多个模型进行bagging不同（普通的bagging方法不同的神经网络时独立的），dropout 共享参数，每个子模型继承父神经网络参数的不同子集，参数共享使得在有限可用的内存下表示指数级数量的模型成为可能， 因此dropout是一种廉价的bagging近似，事实上，Dropout不仅仅是训练一个Bagging的集成模型，而是一种特殊的共享隐藏单元的集成模型，意味着无论其他隐藏单元是否在模型中，每个隐藏单元都必须能够表现练好，相比独立模型集成，可以获得更好的泛化误差的提升。

Dropout强大的另一个原因来自施加到隐藏单元的掩码噪声，破坏提取的特征而不是原始值，让破坏的过程充分利用该模型迄今获得的关于输入分布的所有知识。

集成预测的是由不同分别的算术平均值给出的：$\frac{1}{k}\sum_{i=1}^{k} p^i(y,x)$

dropout 情况下。所有子网络算术平均值给出：$\sum_{\mu} p(\mu)p(y|x,\mu)$,其中$p(\mu)$是训练时采用$\mu$的概率分布，包含指数级的项，无法精确计算。

通常计算采用 **权重比例推断规则** ，有两种做法：1、将某一单元i输出的权重乘以该单元可能出现的概率，保证得到从该单元输出正确的期望值。2、在训练期间将单元的状态除以该单元出现的概率（上面的实现中，采用方法2），目标都是确保在测试时一个单元的期望总输入与在训练时该单元的期望总输入大致是相同的。权重比例推断规则对具有非线性的深度模型仅仅是一个近似，但在实践中效果很好。

实践上dropout相比于其他正则化方法更有效，也可以与其他形式的正则化合并，得到进一步提升。

dropout训练非常快，只需要添加一些节点输出与mask的乘法。

Dropout的另一个优点是不怎么限制适用的模型和训练过程，可以在任何使用分布式表示且可以用随机梯度下降训练的模型上都表现良好。包括FC,CNN,RNN等。

Batch norm(批标准化)

Batch norm（批标准化）可以有效解决covariate shift 的问题，可以让大型神经网络训练速度加快很多倍，同时收敛后的分类准确率也可以得到大幅提高。这里不具体介绍batch_norm的原理，只尝试说明，batch_norm提升泛化能力的原理：

在前文集成提升泛化能力的数学基础中说明了为什么增加模型多样性，可以有效提高模型泛化误差，事实上增加数据的多样性也（比如batch Norm等）同样可以有效提高模型泛化误差，这里就不再证明。

batch_norm 采用mini_batch 方法进行训练，利用每一个batch进行训练时，计算的均值和方差，都是 mini_batch 的均值和方差，因此与全部数据的均值和偏差有差别，在不同层施加执行batch norm，相当于多次给数据增加了不同程度的扰动，增加数据的多样性，使得模型不过分依赖任何某一数据，从而显著提升泛化能力。

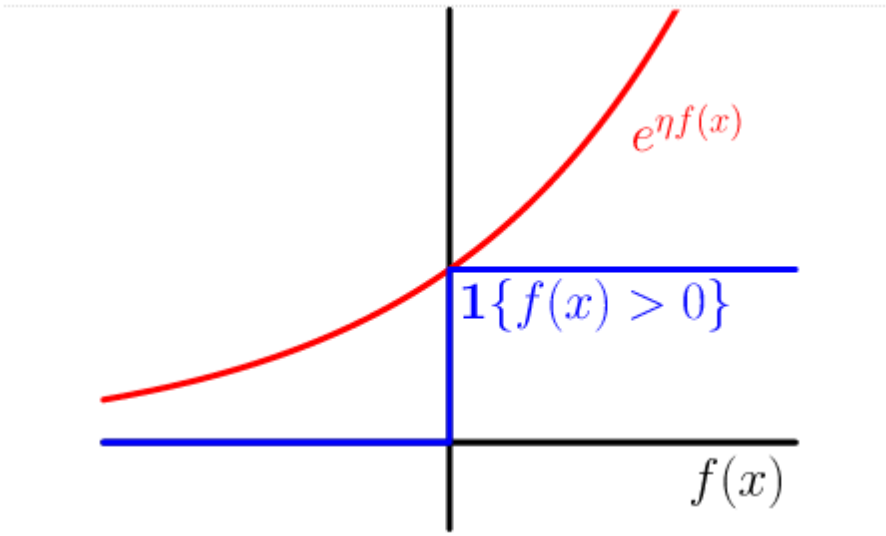
事实上，也可以通过训练多个模型然后进行平均，也可以显著提升泛化能力。另外，由于深度网络，有时中间结果的结果也不错，将其输出，与最终输出进行不同程度的加权集成，也可以显著提升泛化能力，比如Google Inception Net 就采用了这种机制。

Hoeffding 不等式证明

Hoeffding不等式完整描述如下：

定理：设相互独立的随机变量$\xi_1, \dots \xi_N$满足$\xi_i \in [a,b], i = 1, 2...., N$,记$\tilde{\xi} = \frac{1}{N}\sum_{i=1}^{N}\xi_i$,则对于任意的$\epsilon > 0$,有：

$P(\tilde{\xi} - E(\xi) > \epsilon) \leq exp(-2\frac{N\epsilon^2}{(b-a)^2})$,二分类问题中a=0,b=1，因此有$P(\tilde{\xi} - E(\xi) > \epsilon) \leq exp(-2N\epsilon^2)$



定义$f(x) = \tilde{\xi} - E(\xi)$

$P(f(x) > \epsilon) = E[1(f(x) - \epsilon) > 0)$

由指数函数的性质，可知，对于任意的$\eta > 0$，有$e^{\eta(f(x)-\epsilon)} > ((f(x) - \epsilon) > 0)$,(这里另$f(x) = f(x) - \epsilon$)从而有：

$P(f(x) > \epsilon) = E[1(f(x) - \epsilon) > 0) \leq E[exp^{\eta(f(x)-\epsilon)}]$

根据上式有：

$P(\tilde{\xi} - E(\xi) > \epsilon) = P(N\tilde{\xi} - NE(\xi) > N\epsilon) \leq E[exp^{\eta(N\tilde{\xi} - NE[\xi] - N\epsilon)}]] = exp^{-N\eta\epsilon}\prod_{i=1}^{N}E[exp^{\eta(\xi_i - E[\xi])}]$

最后一步的证明是由$\xi_1, \xi_2 \dots \xi_N$相互独立得到的。

设Z是一个随机变量，且$a \leq Z \leq b$,对于任意的实数η，有：

$E[exp^{\eta Z}] \leq \frac{b-E[Z]}{b-a}exp^{\eta a} + \frac{E(z)-a}{b-a}exp^{\eta b}$

由于$exp^{\eta x}$为凸函数，$E(Z) = \frac{1}{2}(a+b)$从而有$E(exp^{\eta Z}) \leq \frac{1}{2}(exp^{\eta a} + exp^{\eta b}) = \frac{b-E[Z]}{b-a}exp^{\eta a} + \frac{E(z)-a}{b-a}exp^{\eta b}$

$E[exp^{\eta(\xi_i - E[\xi])} = exp^{-\eta E[\xi]}E[exp^{\eta\xi_i}] \leq exp^{-\eta E[\xi]}(\frac{b-E[\xi]}{b-a}exp^{\eta a} + \frac{E(\xi)-a}{b-a}exp^{\eta b})$

记$p_i = \frac{E(\xi)-a}{b-a}, \eta_i = \eta(b-a)$

上式经过化简可以得到：

$E[exp^{\eta(\xi_i - E[\xi])} \leq exp^{-\eta_i p_i + log^{1-p_i + p_i exp^{\eta_i}}}$

定义$L(\eta_i) = -\eta_i p_i + log^{1-p_i+p_i exp^{\eta_i}}$

求导可以得到

$L'(\eta_i) = -p_i + \frac{p_i}{(1-p_i)exp^{-\eta_i}+p_i}$

$L''(\eta_i) = \frac{p_i(1-p_i)exp^{-\eta_i}}{((1-p_i)exp^{-\eta_i}+p_i)^2} \leq \frac{\frac{1}{4}((1-p_i)exp^{-\eta_i}+p_i)^2}{((1-p_i)exp^{-\eta_i}+p_i)^2} = \frac{1}{4}$

上式证明用到了$2ab \leq a^2 + b^2$，另$a = p_i, b = (1-p_i)exp^{-\eta_i}$,带入即可证明

将$L(\eta_i)$在$\eta_i = 0$处泰勒展开，

$L(\eta_i) = L(0) + L'(0)\eta_i + \frac{1}{2}L''(\xi)\eta_i^2 \leq \frac{1}{8}\eta_i^2 = \frac{1}{8}\eta^2(b-a)^2$

将$\eta_i = 0$带入，可得$L(0) = 0$，与$L'(0) = 0$

从而根据可以得到

$P(\tilde{\xi} - E(\xi) > \epsilon) \leq exp^{-N\eta\epsilon}\prod_{i=1}^{N}(\frac{N}{8}\eta^2(b-a)^2 - N\eta\epsilon)$

上式对于任意的$\eta > 0$都成立，将不等式右边配方，可以得到$\eta = \frac{4\epsilon}{(b-a)^2}$时，不等式右边取得最小值，从而有：

$P(\tilde{\xi} - E(\xi) > \epsilon) \leq exp(-2N\epsilon^2)$

证明完毕。

上述VC维内容是本人自学台大林轩田机器学习基石和斯坦福吾恩达机器学习课程的总结，偏差-方差实例说明来自台大李宏毅机器学习课程。正则化实例部分主要是吴恩达深度学习课程总结。Hoeffding 不等式证明根据网上一篇博客：http://freemind.pluskid.org/slt/vc-theory-hoeffding-inequality/

VC维理论非常复杂，作为自学者，肯定有很多不够严谨的论述，因此如果发现任何论述不到位或者错误的地方，欢迎批评指正，谢谢。