

## P4 Basic Concept

#机器学习

---

Error due to bias 偏差

Error due to variance 方差

如果能够诊断你的Error来源就可以提高你改进Model的效率

---

## Estimator 估算

$y^H = f^H(x)$  理论上（理想）最佳的function 记为  $f^H$

$y^H$  是我们不知道的，所以我们只能 From training data,

We find  $f^*$  (根据training data找到)

$f^*$  is an estimator of  $f^H$  ( $f^*$  是理想函数的一种估计，估算，推测)

---

## 证明 N越大 误差越小

Bias and Variance of Estimator

Estimate the mean of a variable x

(假设目前有一个变量X)

assume the mean of x is  $\mu$  ( $\mu$ )

假设这个变量的平均值是  $\mu$  (前面带长线)

assume the 均方差 of x is  $\sigma^2$

假设他的差异是  $\sigma^2$  平方

Estimate of mean  $\mu$

(我们要如何估测  $\mu$ )

Sample N point:  $\{x^1, x^2, \dots\}$

取样N个点

$$m = \frac{1}{N} \sum_n x^n$$

$m$  是估测出来的平均值 一定是 不等于  $u$

$$E[m] = E\left[\frac{1}{N} \sum_n x^n\right] = \frac{1}{N} \sum_n E[x^n]$$

$m$  的期望值 等于  $u$

就好像是说打靶的时候

准星是瞄准  $u$  的

但是因为种种原因，你会散落在你瞄准位置的周围

那么这个散布在周围，散步的多远呢？

$$\text{Var}[m] = \frac{\sigma^2}{N}$$

取决于  $m$  的 variance 方差

Variance 的值取决于你取了多少个点

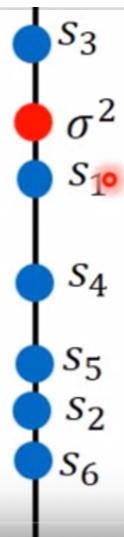
- Estimator of variance  $\sigma^2$ 
  - Sample N points:  $\{x^1, x^2, \dots, x^N\}$

$$m = \frac{1}{N} \sum_n x^n \quad s^2 = \frac{1}{N} \sum_n (x^n - m)^2$$

**Biased estimator**

25

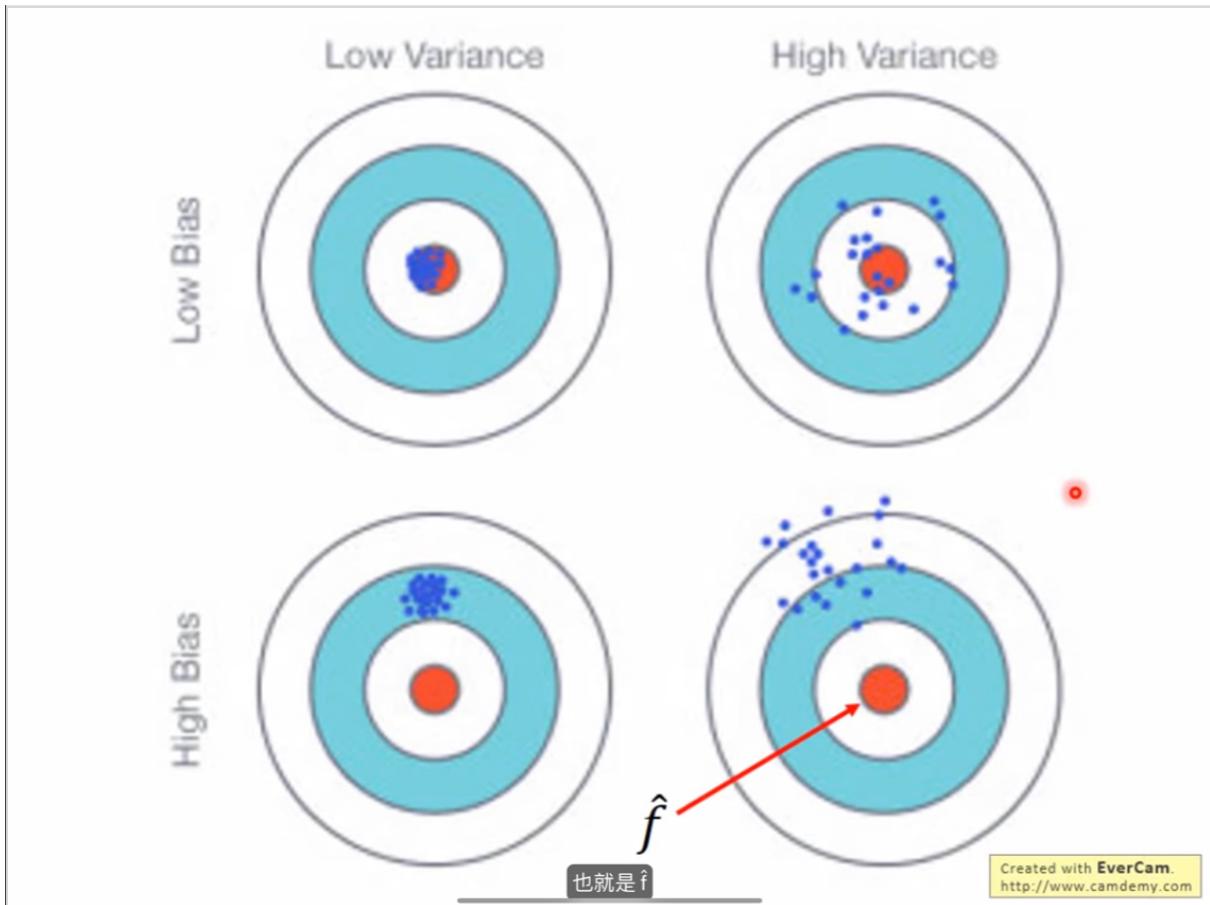
$$E[S^2] = \frac{N-1}{N}$$



Biased estimator

主要是证明了  $N$  越大 这个误差就越小，这个部分需要概率论的知识（所以先放着）

Error 的 bias 和 variance 区别



我们现在要估测靶的中心 也就是  $F^H$

这个是目标

收集了一些数据 找到了一些  $F^*$  如图4

这个  $F^*$  跟红心的 Error 取决于两件事

1. 你瞄准 (Estimate) 的位置在哪里，  
怎么样知道瞄准的是不是靶心呢  
我们需要算出来  $F^*$  的期望值

如果 期望值相等 就是你的 瞄准 跟 靶心是在一起的

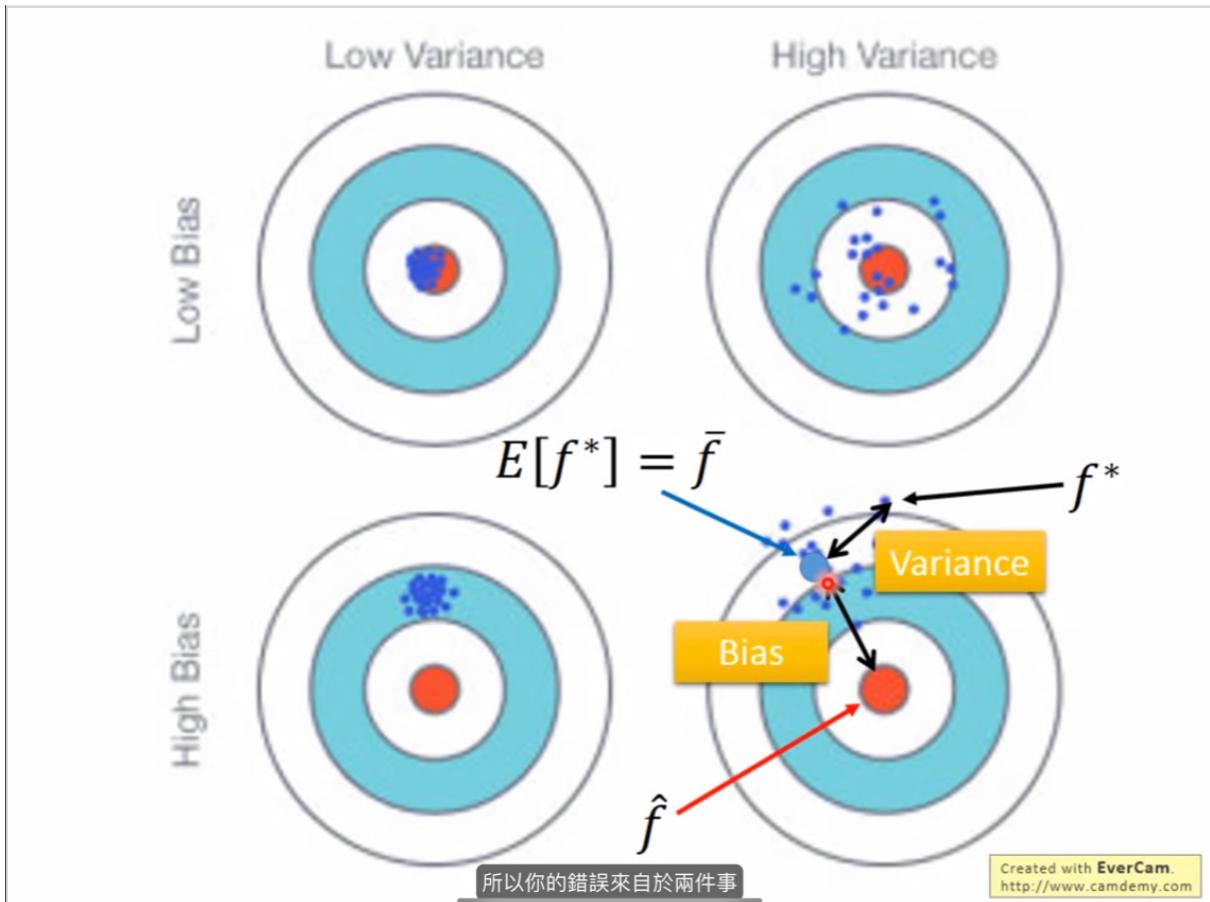
如果有差距，就说明你瞄准的不是靶心

你根本就没有瞄准

这个就叫做 bias

2. 你瞄准了这个位置，但是子弹射出去有偏差

这个就叫做 variance



所以图4 既有没有瞄准 (bias) , 和子弹有偏差 (variance)

最理想的状态是既没有bias 也没有 variance

也有可能是图三 就是 bias很大，但是 variance很小，每次F\*都很像，但是集中错在一个位置

也有可能是图二 就是bias很小，但是variance很大

## Variance

Parallel Universes (平行宇宙)

In all the universes , we are collecting (catching) 10 Pokemon's as training data to find F\*  
在很多个平行宇宙，每个宇宙都不一样，但是我们都抓10只宝可梦做分析

In different universes ,we use the same model, but obtain different F\*  
在不同宇宙我们用同一个model，假设我们用

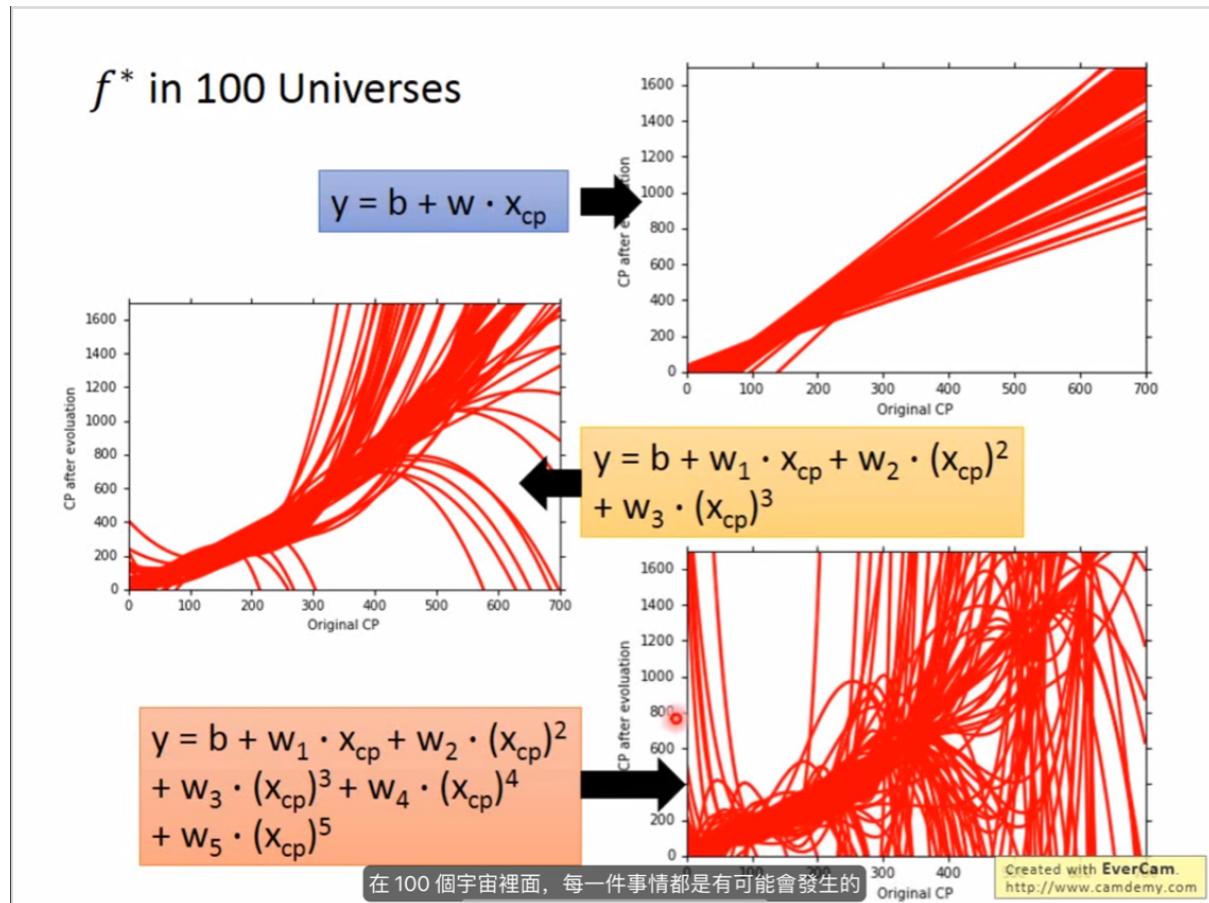
$$Y = b + w * x_{cp}$$

这个Model

但是给出不同的dataset (数据集) 这样找出来的function就不一样

$F^*$  in 100 Universes

(世界上并没有平行宇宙, 你可以做100实验, 每次都抓十只不同的宝可梦就可以)



我们会发现当model趋于复杂对于输入的变化就越敏感

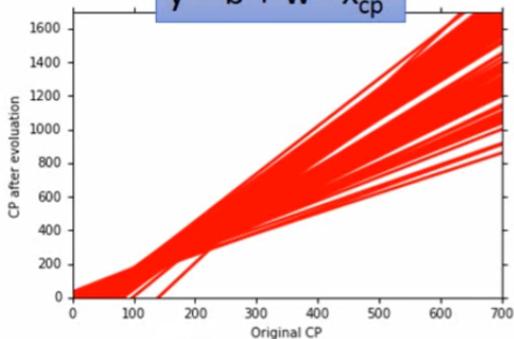
如果我们看这个model之间的variance, 以100次实验来说, 简单的model就是只考虑一次的model他是比较集中的,

如果是考虑5次的话, 就是会散的非常开的

也就是说用比较简单的model他的variance是比较小的 (variance就是方差也就是发散程度)  
相反, 用复杂的model, variance会偏大

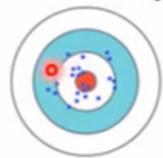
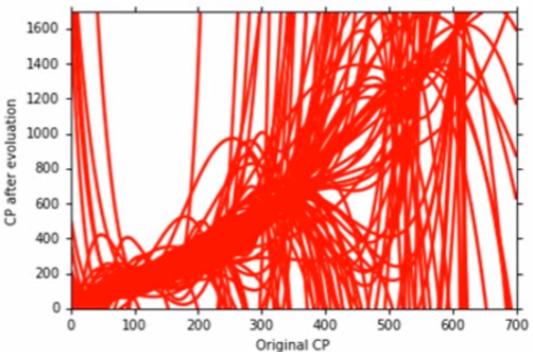
# Variance

$$y = b + w \cdot x_{cp}$$



Small  
Variance

$$\begin{aligned}y &= b + w_1 \cdot x_{cp} + w_2 \cdot (x_{cp})^2 \\&+ w_3 \cdot (x_{cp})^3 + w_4 \cdot (x_{cp})^4 \\&+ w_5 \cdot (x_{cp})^5\end{aligned}$$



Large  
Variance

它的散佈就很開，就像這邊藍色的點一樣

Created with EverCam.  
<http://www.camdemmy.com>

为什么复杂的model就散的开，而简单的model就不会呢？

Simpler model is less influenced by the sampled data

因为越简单的模型会被初始值影响的越小

比较不会受你的data影响

思考一个极端的例子  $F(X) = C$

这个model就完全不受初始值的影响

## Bias

If we average all the  $F^*$ , is it closed to  $F^H$

假设我们有很多的  $F^*$  我们求他的平均值是否靠近  $F^H$

Large Bias

就是把所有的  $F^*$  平均起来跟  $F^H$  有一段距离

Small

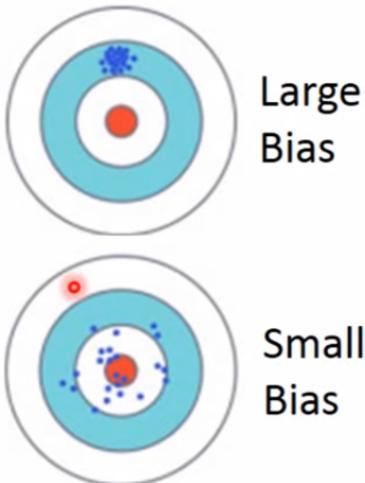
就是把所有的  $F^*$  平均起来跟  $F^H$  没有距离

这里我们不管单个的 $f^*$ 与 $\hat{f}$ 的距离，我们只关心平均值

## Bias

$$E[f^*] = \bar{f}$$

- Bias: If we average all the  $f^*$ , is it close to  $\hat{f}$



它分散得多开我們不管，你要找它的平均值

Created with EverCam.  
<http://www.camdemmy.com>

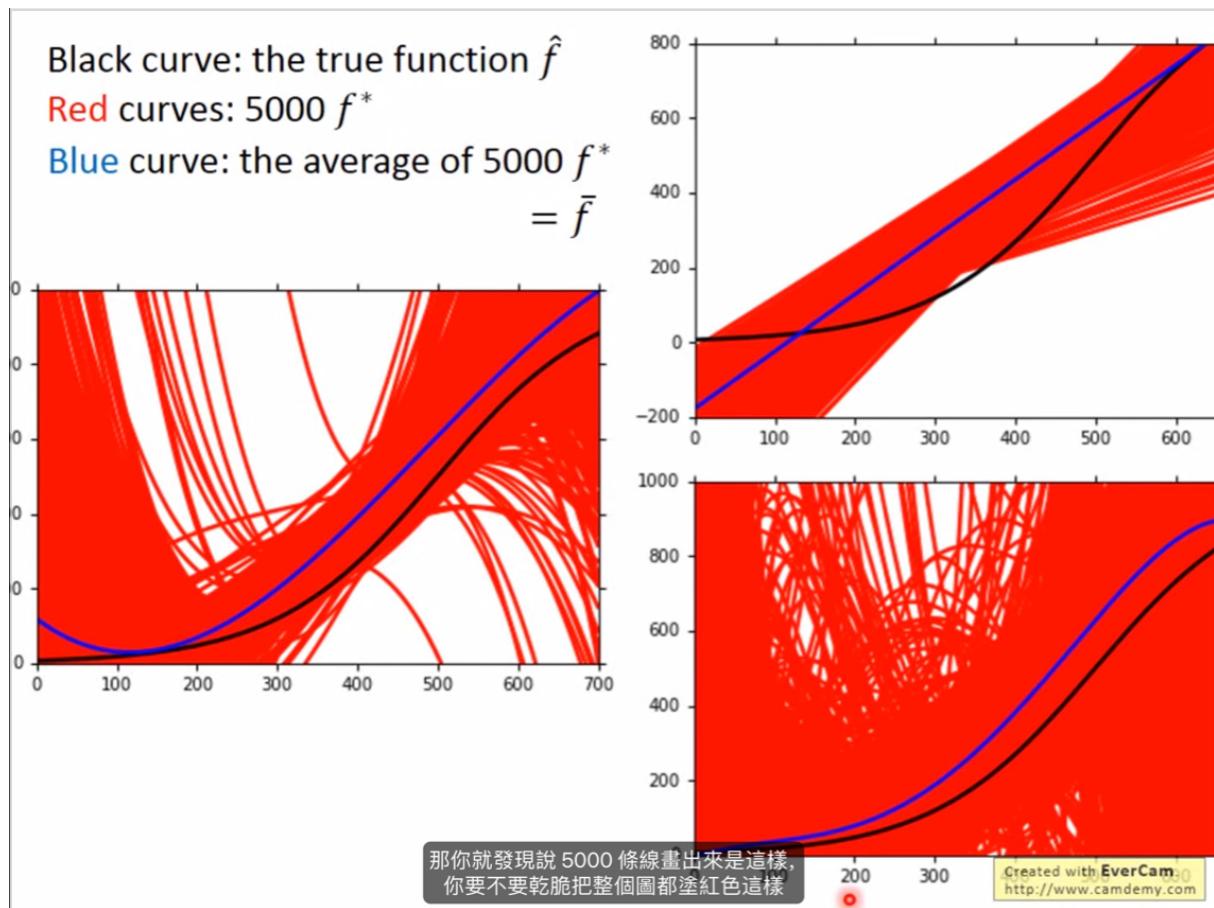
如果我们想要量不同function之间的bias之间的Bias有多大

其实根本没有办法量

因为我们不知道 $F^H$

Assume the  $F^H$

所以我们只能假设一个 $F^H$



黑色的是理想的 $F^H$

红色是5000次实验的 $F^*$

蓝色是平均线

左边是三次

右上是一次

右下五次

我们会发现就是 三次的平均曲线要比一次式子更加贴合理想曲线

而五次式子要比三次式子更加贴合曲线，非常贴合理想曲线！！

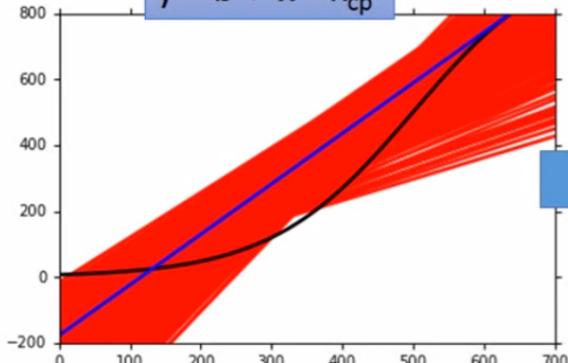
说明 五次Bias是要比三次的小， 三次的bias要比二次的小

所以一个比较简单model 有比较小的variance， 有比较大的bias

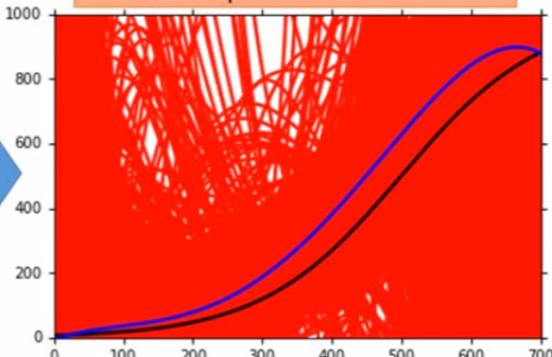
反而一个比较复杂model 有比较大的variance， 有比较小的bias

## Bias

$$y = b + w \cdot x_{cp}$$



$$\begin{aligned}y = b + w_1 \cdot x_{cp} + w_2 \cdot (x_{cp})^2 \\+ w_3 \cdot (x_{cp})^3 + w_4 \cdot (x_{cp})^4 \\+ w_5 \cdot (x_{cp})^5\end{aligned}$$



model

Large  
Bias

我們說，我們的 model 就是一個 function set 對不對



Small  
Bias

Created with EverCam.  
<http://www.camdemyc.com>

### 直观解释：

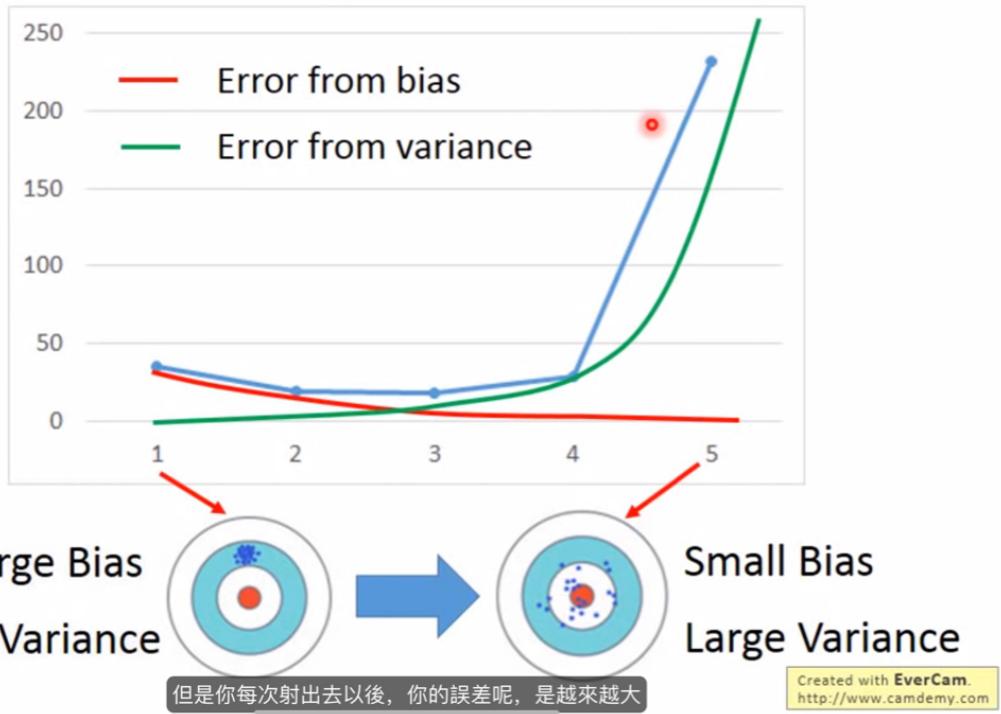
我们的model其实就是一个function Set

当你设定好一个model，你的函数就只能从function Set 里面选出来。

一个简单的的model，函数就比较小，可能根本就没有包含target的model，那无论你怎么选点，平均起来都不会是target，因为你的model里面根本就不包含target

越复杂的model就能包含越多的function，你的model的function space就比较大，就可能包含你的target，只是没有办法找到target在哪里，但是平均起来就可以获得比较准的值，比如五次的model 就包含一次的model（让其余项都等于 0 ）

# Bias v.s. Variance



从左向右

Bias逐渐下降

Variance逐渐上升

所以当这两项同时被考虑的时候，我们需要找到一个平衡点，能够让bias和variance加起来保持最小

如果你的variance很大bias很小，就是overfitting

如果你的bias很大， variance很小，就是underfitting

要明确知道你的问题是Bias大还是Variance大

如果你的model没有办法fit 你的training data就说明 bias大

代表说你的model跟正确model有一段距离，就是underfitting

如果你的model适合train data，但是你在testing data得到一个大的error，你的model就会有一个大的variance，这就是overfitting！

如果是bias大如何处理：

redesign your model 重新设计你的model

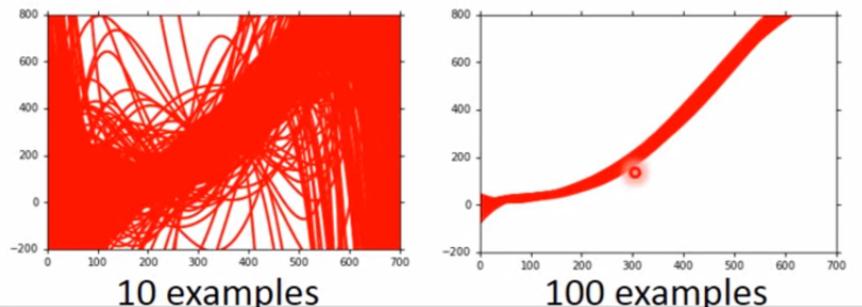
增加更多特性

让你的model更加复杂!

## 如果是variance大:

More Data 增加你的data数据

- More data



增加data数量是一种很有效控制variance的数量

十分有效，万用

不会伤害你的bias

很多时候你不能收集更多数据

有时候也可以手动制造一些data，不如说在手写数字的时候

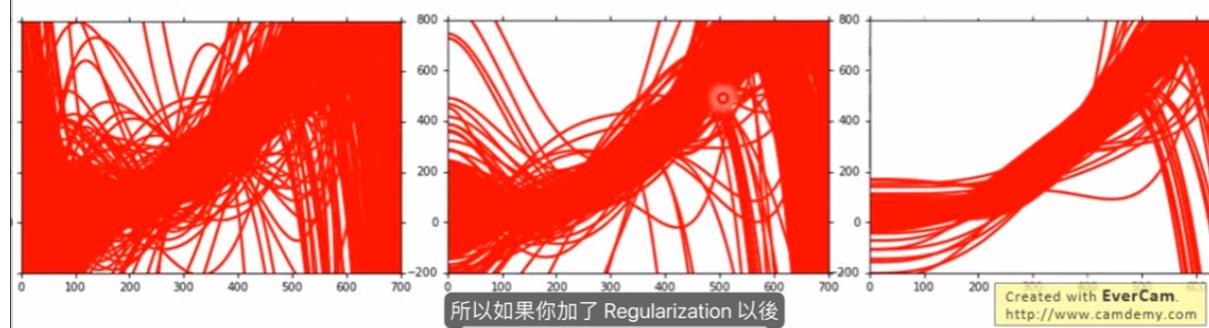
可以自己写一些数字

或者对图片进行旋转

## Regularization正则化

这个会在loss函数里面新加一项，会希望你的参数越小越好，你的函数越平滑越好，前面有一个参数，控制平滑程度

- Regularization



但是平滑化你的曲线会伤害你的bias，有可能会增加bias

所以要调整weight (权重)

所以现在有一个问题，就是我们有很多个model可以选择，有很多参数可以调整

通常我们是在bias和variance之间做一些平衡 (trade-off)

我们希望找一个model， Bias和Variance都能够小，这两个合起来给我最小的testing data的error

以下的事情是不应该去做的：

你有training set 和 testing set

Model 1

Model 2

Model 3

你应该选哪一个Model，你就分别用所有model去找一个Best function，接下来把它带到Test data

就能选出哪个model最好

但是有一个问题，就是这个testing data 是你自己的testing set，是你拿来衡量你model好坏的testing set

你并没有真正的testing set (考试的时候)，就是你的testing set跟考试验证的testing set不完全一样，所以bias也会有差距

---

HomeWork:

Training Set: 你有的Training Date

Puliuc Testing Set: 你有的Testing Set

Private Testing Set:

当你上传到验证网站后，你只能看到你的Public set的分数，而看不到Private Set的分数，  
Private Set的分数要到截止日期后才能看到。

而且一般 Private Set的bias通常要大于public Set，有可能你还是没有通过

所以说Public Set的结果是不可靠的

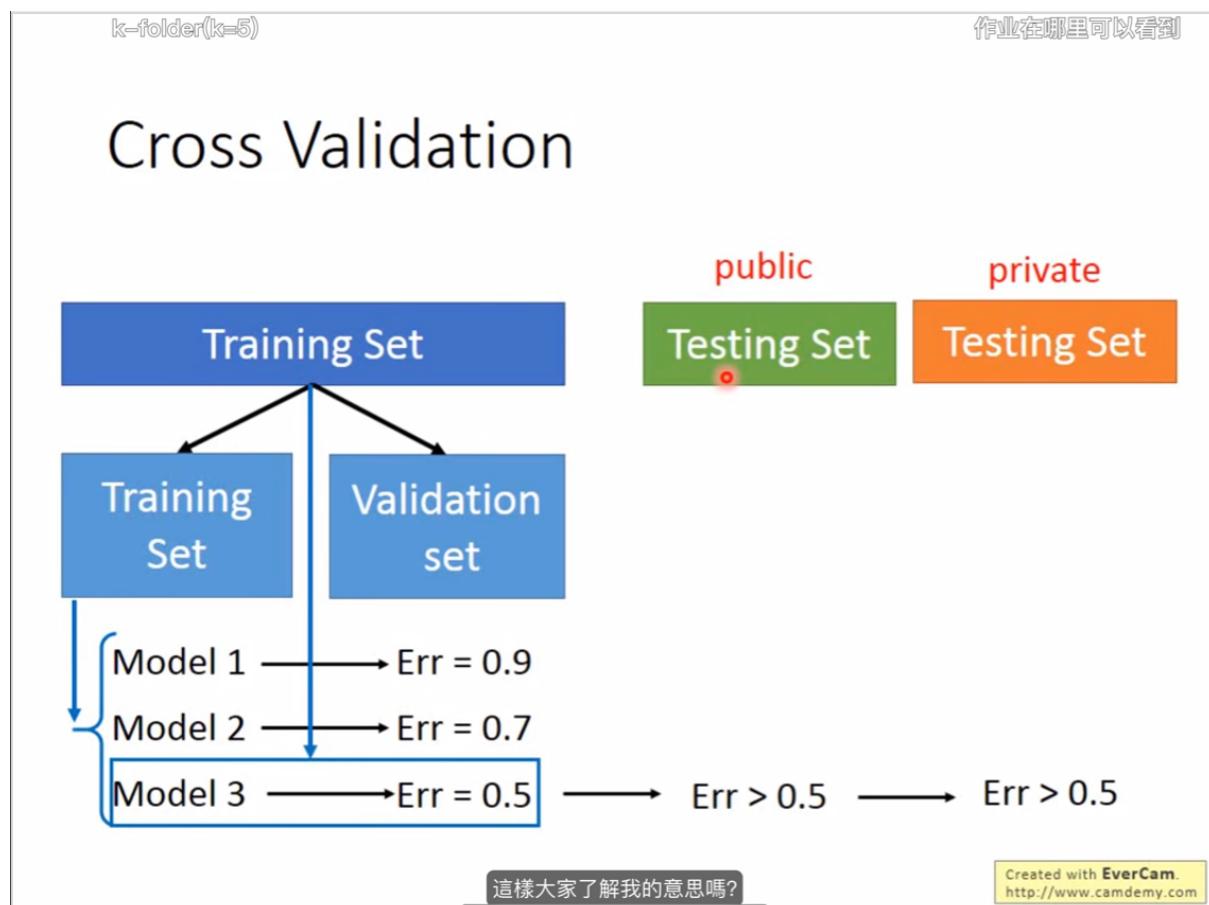
**应该怎么做：**

要把 Training Set分成两组

一组是 Training Set (训练集)

一组是 Validation Set (选Model)

然后训练你的model在小的选出你最好的model，用所有的TrainingSet在训练一次提交到你的testing Set，这样你的Public Set Err才能比较好的反应你的Private Set Err



如果你利用Public Set上的结果回头去改你的Training model又把Public Testing Data 的bios考虑进去了

这样就会变成你在Public Testing Set的成绩无法反应你在Private Testing Set的成绩

Public Set 并不是最终的结果

Public Set是假的！

如果我分不好

就是无法分号 Training Set和Validation Set

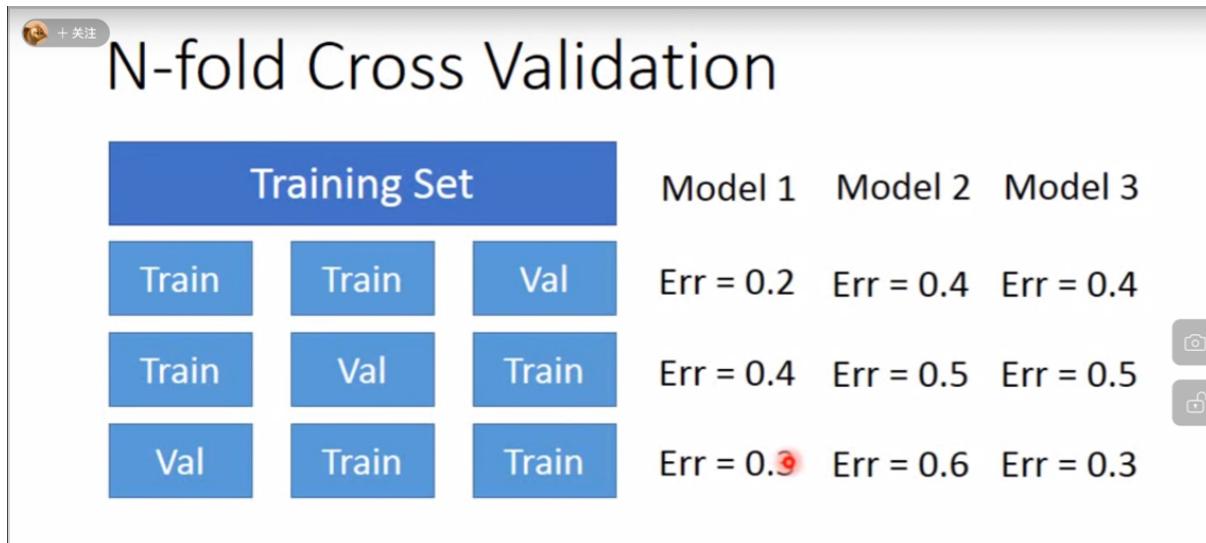
那么可以做

N-fold Cross Validation N折交叉验证

就是分很多次Testing data 和 Validation Test

比如说你把你的training set 分成三份

每一次都拿一份做validation，另外两份做Training data



可以算一下每个model的Ave Err

在用最好的model用整个数据集训练最后提交到你的Public Test