Bias & Variance

在模型训练种，有 2 种 Error，那就是$Bias$和 $Variance$。

基础原理

$$Mean\ of\ x \rightarrow \mu$$

$$Variance\ of\ x \rightarrow \sigma^2$$

要找 $mean\ \mu$，首先需要$Sample\ N\ Points: \{x^1, x^2, \dots, x^n\}$，然后再计算$mean\ m$。

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

算出来的 m 是和 μ 不一样的，因为 μ 代表的是所有 Data，而 m 代表的是我们有的 Data，每一次算出来的 m 都是不一样的。这时候我们可以做的就是计算 m 的期望值 $E[m]$。

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

Sample 的 Data 有多散开或集中，取决于 m 的 $Variance$。

$$Var[m] = \frac{\sigma^2}{N} \rightarrow Variance\ Depends\ on\ the\ Number\ of\ Samples$$

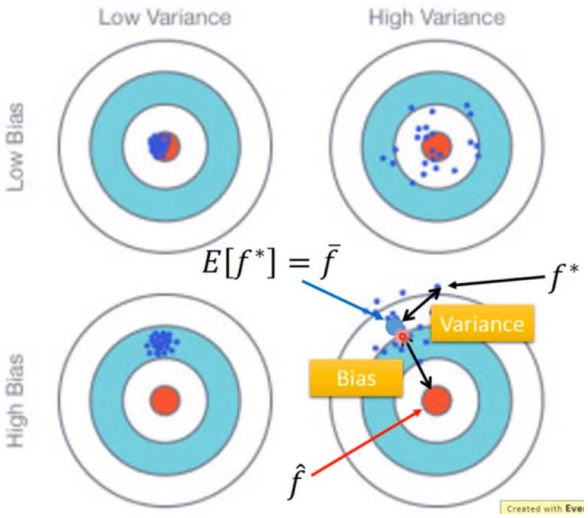
$$s^2 = \frac{1}{N}\sum_n (x^n - m)^2$$

$$Variance的期望值 \rightarrow E[s^2] = \frac{N-1}{N}\sigma^2$$

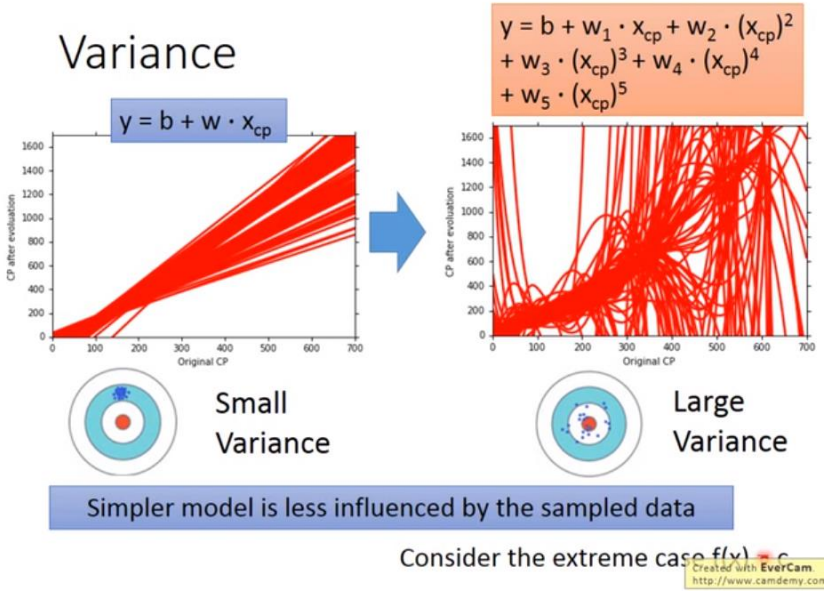
$E[s^2]$是 s^2 的期望值，它是 Bias 的，算出来不会等于σ^2，而是$\frac{N-1}{N}$。通常算出来的 s^2 是比 σ^2 小。如果 N 越来越大，算出来的 σ^2 和 s^2 的差距就会越来越小。

Model Training 模型训练

$$\hat{f} \rightarrow Best\ Function$$

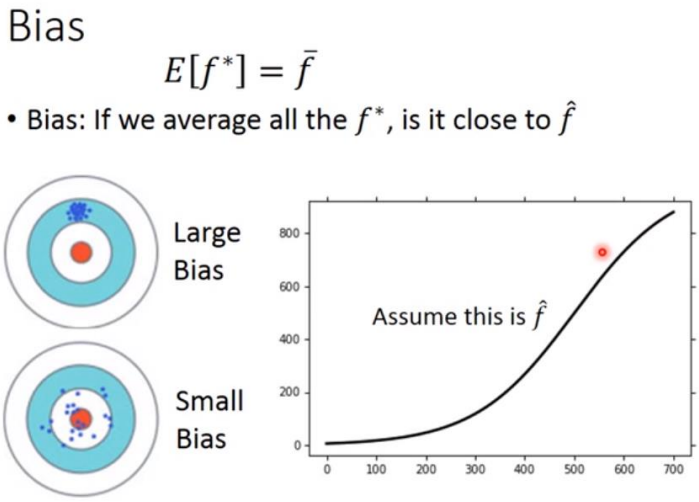


在 $\hat{f}$ 与算出的 f^* 的期望值 $E[f^*] = \bar{f}$ 之间的差别就是 Bias。而 f^* 与 $\bar{f}$ 之间的扩散程度就是 $Variance$。

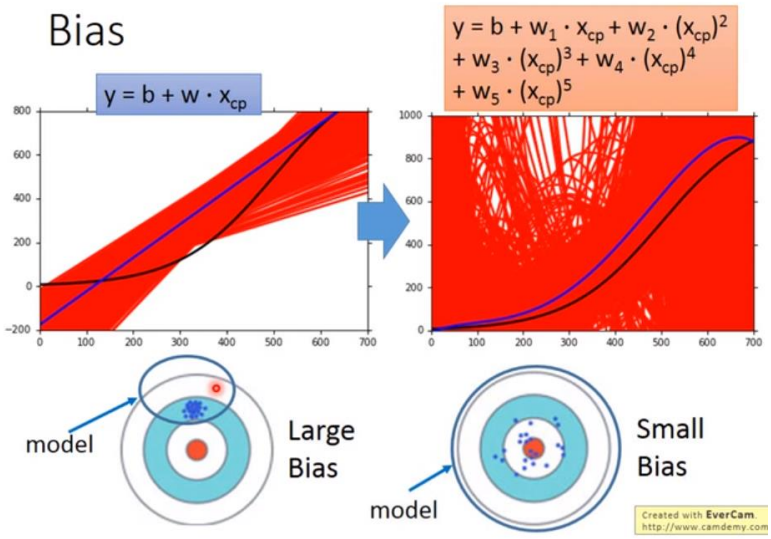


上图是 100 个不同的模型算出来的的结果。当 Function 的维度越低，算出来的的结果是比较集中 (Small Variance)，而 Function 的维度越高，算出来的结果就是比较杂乱 (Large Variance)。

Bias 其实就是对所有的f^* 取平均值然后计算与 $\hat{f}$ 的差别有多大。

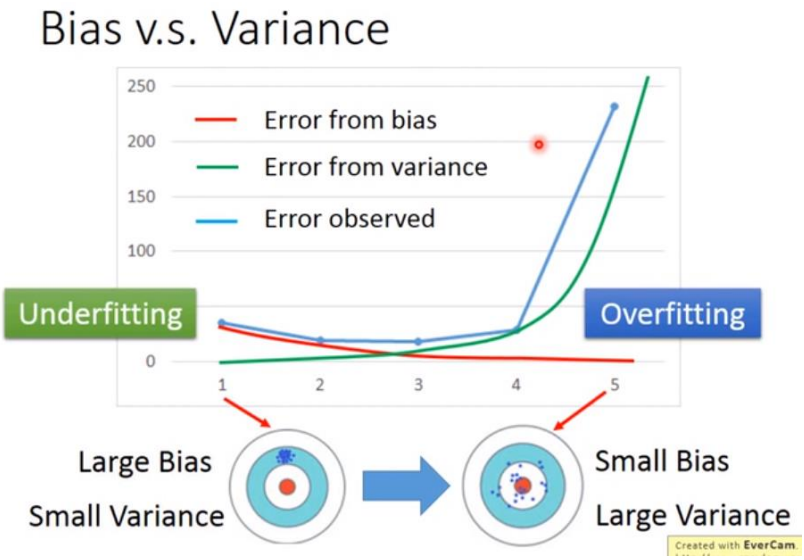


当所有的f^* 很集中，但是与 $\hat{f}$ 的差别很大是，就是 Large Bias。而当所有的 f^* 很不集中，但是离 $\hat{f}$ 的差距很小时，就是 Small Bias。在 Bias 不在乎 f^* 集不集中，只在乎里 $\hat{f}$ 的距离有多大。



上图是对跑了 5000 的 f^* 选平均值 (蓝色线) 之后再与提前设定的 $\hat{f}$ (黑色线) 做出对比。低维的 Function 算出来的模型的 Bias 比较大，但是 Variance 比较小。而高维的 Function 算出来的模型 Bias 比较小，但是 Variance 很大。

可以看出高纬 Function 算出来的 f^* 平均值是比较接近 $\hat{f}$ 的。这是因为$y = b + wx$ 是直线的 Function，不管怎么算它就只能是直线，而当 $\hat{f}$ 它不是直线的时候，结果就会很差。而高维的 Function 可以算出比较不一样的曲线。



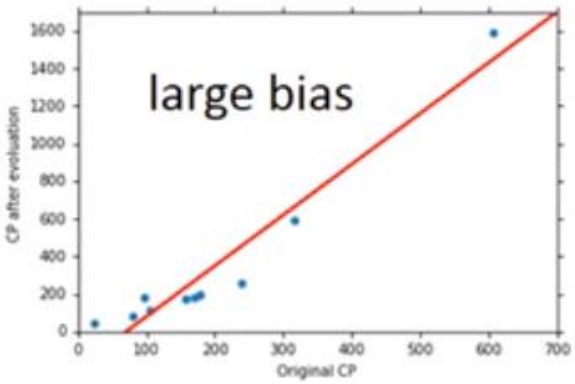
$$Large\ Bias \rightarrow Underfitting$$

$$Large\ Variance \rightarrow Overfitting$$

上图蓝色线是 Error from Bias 和 Error from Variance 的平均值。通常的情况是维度越低，训练出来的模型就是 Large Bias 和 Small Variance。而维度越高，训练出来的模型就是 Small Bias 和 Large Variance。

Underfitting (Large Bias)

$$Large\ Bias \rightarrow Underfitting \rightarrow 模型不能给出正确的Prediction\ (Training\ 和\ Testing)$$



Large Bias 的时候，加入更多的 Data 来训练是没办法训练出好的模型。

解决方法:

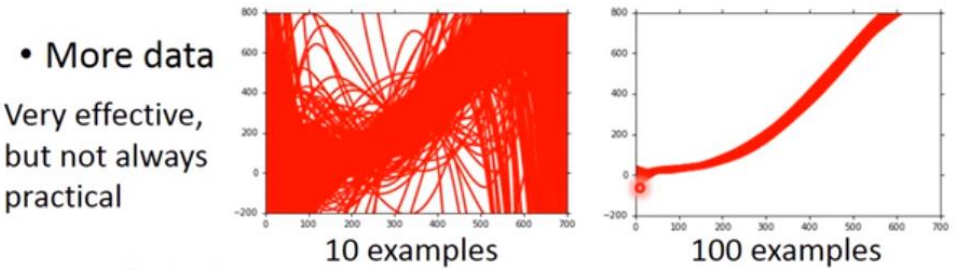
I. Add more features (inputs)
II. More Complex Model 可以用更高维度的 Function ($y = b + wx + w(x)^2 + \cdots + w(x)^n$)。

Overfitting (Large Variance)

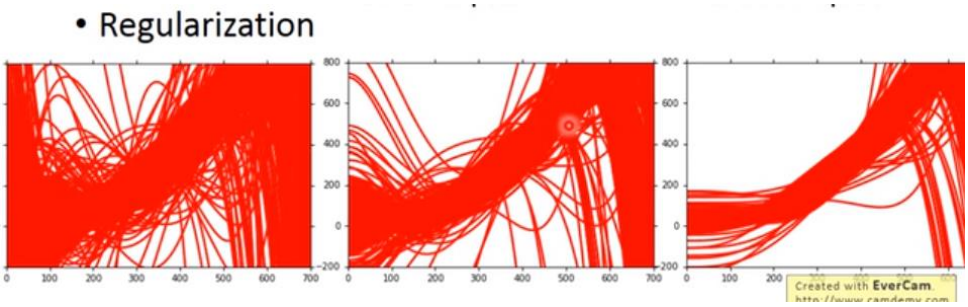
$$Large\ Variance \rightarrow Overfitting \rightarrow Training\ Loss很小但是在Testing\ Data\ 的Accuracy很低$$

解决方法:

I. 增加更多的 Data 可以解决 Overfitting 的问题，但是现实中，要找到更多的 Data 是比较难的事情



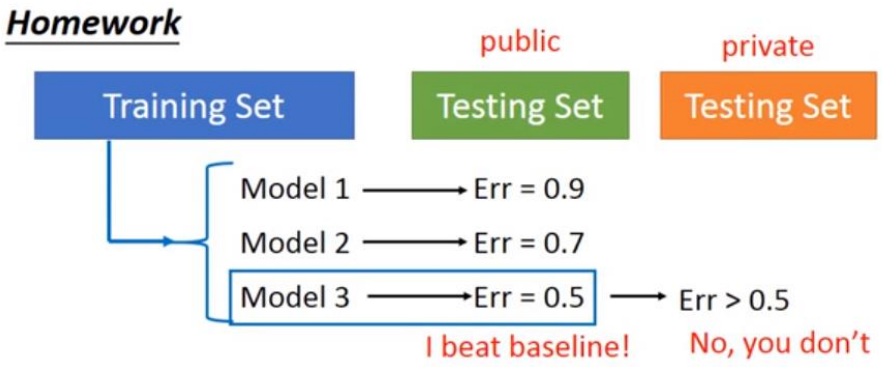
II. 使用 Regularization 也可以解决 Overfitting 的问题。



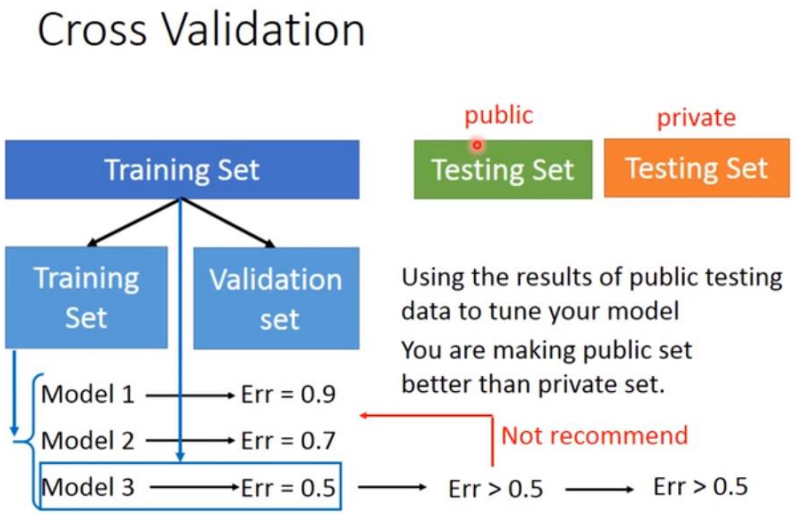
Regularization 里，λ 的值越大，算出来的 f^* 就会越平滑。使用 Regularization 它是会伤害模型的 Bias，导致模型的表现不好 (Low Accuracy)。当使用 Regularization 的时候，需要调整 Regularization 的 λ，在 Bias 和 Variance 之间取得平衡。

Dataset Distribute (数据分配)

通常训练好模型之后，直接跑 Testing Set 之后选出最好的模型。当这个模型使用在真正的 Dataset 上，通常 Error 都会比较高。

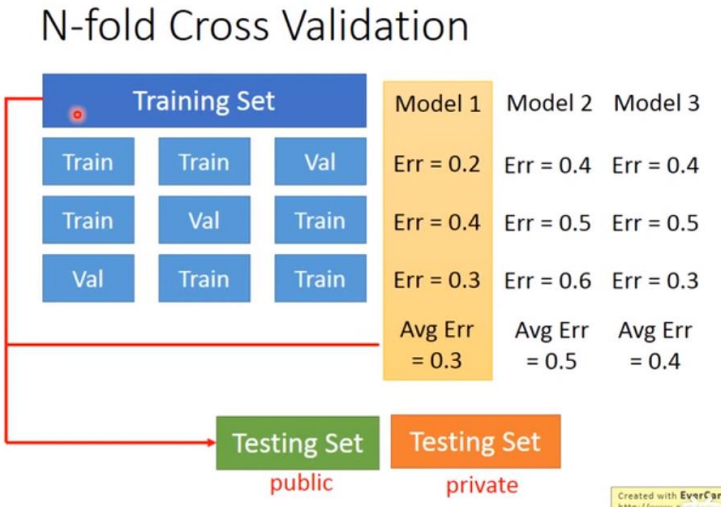


训练时，通常使用了 Cross Validation 的方法。在原来的 Training Set 里，分成两组，一组 Training Set 是用来训练模型，另一组是 Validation Set 是用来选出最好的模型。当选出最好的模型之后，通常会把 Validation Set 也加入选好的模型里一起训练。



使用这个方法的好处就是可以确保当训练和选出的模型在真正的 Data 上面跑，误差不会太大。有些人看到模型在 Testing Set 的 Accuracy 不高，会选择把 Testing Set 也加入到 Training Set 里面，这样子做虽然模型训练好了在 Testing Set 的 Accuracy 会提高，但是对真正的 Data 没有太大的帮助。

还有一种方法是 N-Fold Cross Validation。这个方法是把 Training Set 分成 3 分，2 分 Training Set，1 分 Validation Set。然后进行训练。



训练好后，取 3 次训练的 Average Error，然后选择 Error 最低的模型，再使用整个 Training Set 来训练选好的模型，之后再用 Testing Set 来测试。