

# Chapter 2 Regression

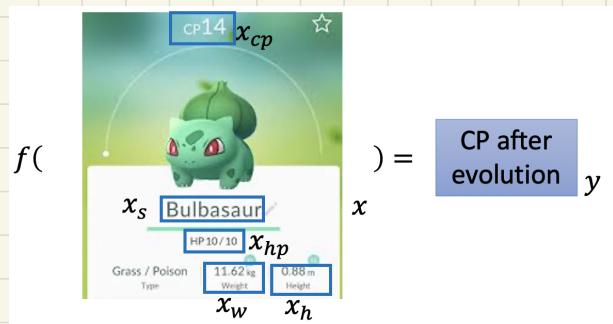
## 1. Example Application

股市预测： $f(\text{历史股市}) = \text{明日道琼斯指数}$

自动驾驶： $f(\text{路况}) = \text{方向盘角度}$

推荐系统： $f(\text{使用者A, 商品B}) = \text{购买的可能性}$

本节使用例子：预测一只宝可梦进化的CP (Combat Power) 值



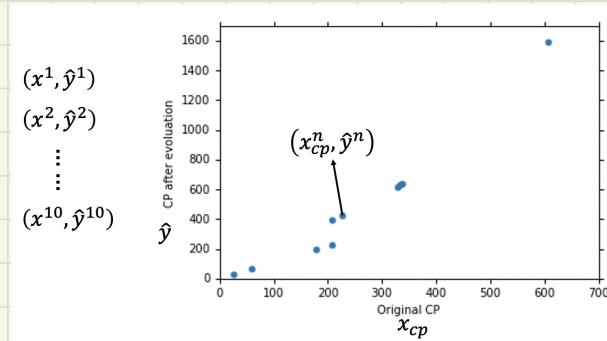
## 2. Model

假设预测模型为 Linear Model:  $y = b + \sum w_i x_i$

$\downarrow \text{bias}$        $\downarrow \text{weights}$        $\xrightarrow{\text{features}}$

## 3. Goodness of Function

①训练数据： $x^i$  表示进化前宝可梦的信息向量， $\hat{y}^i$  表示进化的 CP 值



## ② 损失函数:



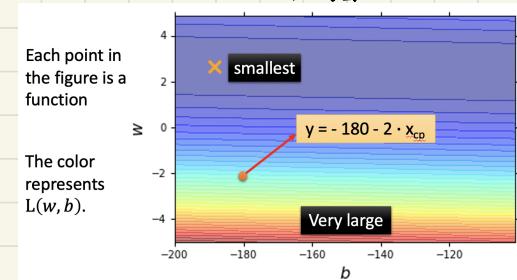
Loss function  $L$ : input - a function output - how bad it is

$$L(f) = L(w, b) = \sum_{n=1}^{10} (\hat{y}^n - (b + w \cdot x_{cp}^n))^2$$

选择最好的模型函数

$$f^* = \arg \min_f L(f)$$

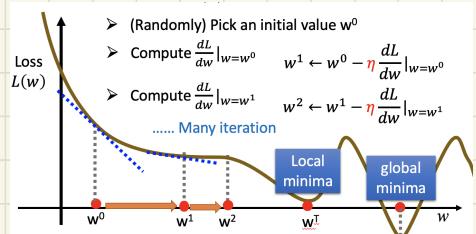
$$w^*, b^* = \arg \min_{w,b} L(w,b)$$



## 4. Gradient Descent

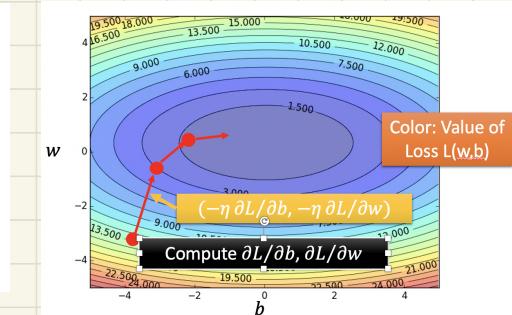
① 对于只有一个参数  $w$  的损失函数  $L(w)$ , 目标为  $w^* = \arg \min_w L(w)$

- 随机初始一个值  $w^0$
- 计算  $\frac{dL}{dw}|_{w=w^0}$ , 更新参数  $w^1 \leftarrow w^0 - \eta \frac{dL}{dw}|_{w=w^0}$
- 计算  $\frac{dL}{dw}|_{w=w^1}$ , 更新参数  $w^2 \leftarrow w^1 - \eta \frac{dL}{dw}|_{w=w^1}$
- 不断迭代, 直至  $\frac{dL}{dw}|_{w=w^n} = 0$ , 找到解  $w^* = w^n$



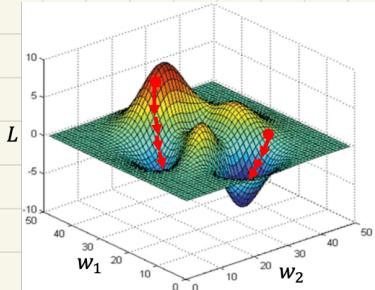
② 对于  $L(w, b)$ , 目标为  $w^*, b^* = \arg \min_{w,b} L(w,b)$

- (Randomly) Pick an initial value  $w^0, b^0$
- Compute  $\frac{\partial L}{\partial w}|_{w=w^0, b=b^0}, \frac{\partial L}{\partial b}|_{w=w^0, b=b^0}$   
 $w^1 \leftarrow w^0 - \eta \frac{\partial L}{\partial w}|_{w=w^0, b=b^0} \quad b^1 \leftarrow b^0 - \eta \frac{\partial L}{\partial b}|_{w=w^0, b=b^0}$
- Compute  $\frac{\partial L}{\partial w}|_{w=w^1, b=b^1}, \frac{\partial L}{\partial b}|_{w=w^1, b=b^1}$   
 $w^2 \leftarrow w^1 - \eta \frac{\partial L}{\partial w}|_{w=w^1, b=b^1} \quad b^2 \leftarrow b^1 - \eta \frac{\partial L}{\partial b}|_{w=w^1, b=b^1}$



### ③ 局部最优与全局最优

参数选择的初始值不同，会找到不同的最小值（全局/局部）



对线性回归不存在局部最小值，因为损失函数是凸函数 (convex)

$$④ \text{理论计算: } L(w, b) = \sum_{n=1}^{10} (\hat{y}^n - (b + w \cdot x_{cp}^n))^2$$

$$\frac{\partial L}{\partial w} = \sum_{n=1}^{10} 2(\hat{y}^n - (b + w \cdot x_{cp}^n))(-x_{cp}^n)$$

$$\frac{\partial L}{\partial b} = \sum_{n=1}^{10} 2(\hat{y}^n - (b + w \cdot x_{cp}^n))(-1)$$

## 5. Overfitting

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

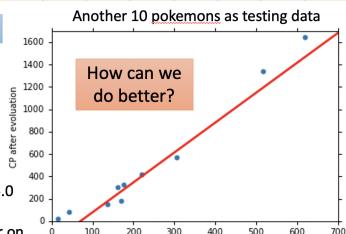
$$b = -188.4$$

$$w = 2.7$$

Average Error on Testing Data

$$= \frac{1}{10} \sum_{n=1}^{10} e^n = 35.0$$

> Average Error on Training Data (31.9)



#### Selecting another Model

$$y = b + w_1 \cdot x_{cp} + w_2 \cdot (x_{cp})^2 + w_3 \cdot (x_{cp})^3$$

#### Best Function

$$b = 6.4, w_1 = 0.66$$

$$w_2 = 4.3 \times 10^{-3}$$

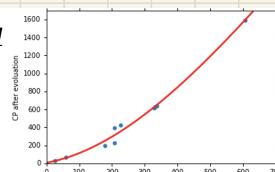
$$w_3 = -1.8 \times 10^{-6}$$

Average Error = 15.3

#### Testing:

Average Error = 18.1

Slightly better.  
How about more complex model?



#### Selecting another Model

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

#### Best Function

$$b = -10.3$$

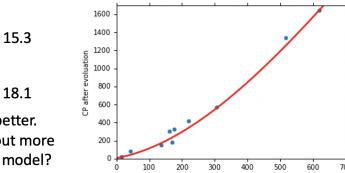
$$w_1 = 1.0, w_2 = 2.7 \times 10^{-3}$$

Average Error = 15.4

#### Testing:

Average Error = 18.4

Better! Could it be even better?



#### Selecting another Model

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

#### Best Function

$$b = 14.9$$

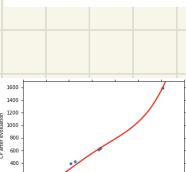
$$w_1 = 0.0, w_2 = 0.0$$

Average Error = 14.9

#### Testing:

Average Error = 28.8

The results become worse ...



#### Selecting another Model

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

#### Best Function

$$b = 12.8$$

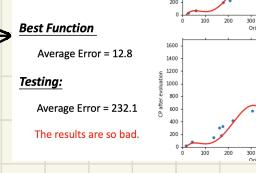
$$w_1 = 0.0, w_2 = 0.0$$

Average Error = 12.8

#### Testing:

Average Error = 232.1

The results are so bad.

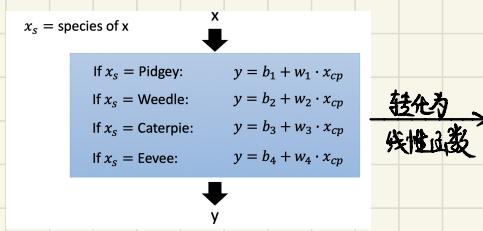


如果一个模型过于复杂，即便其可以在训练集上达到很小的误差，但其在测试集上的表现不一定很好，该现象称为过拟合 (Overfitting)

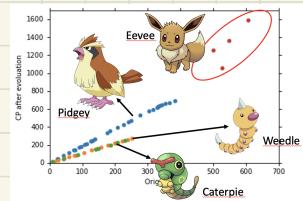
## 5. Redesign Model

① 获取更多的训练数据后会发现物种之间的差异也是影响预测CP值的因素之一

因此，设计更为复杂的模型如下：

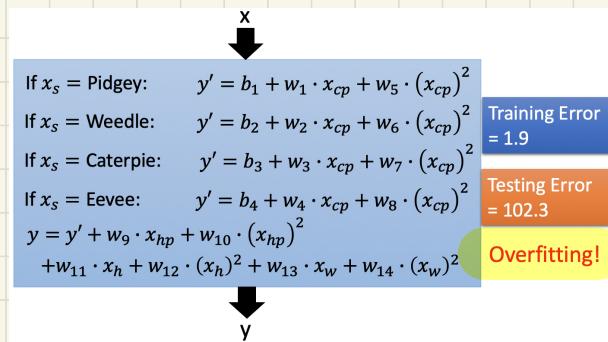


$$y = b_1 \cdot \delta(x_s = \text{Pidgey}) \\ + w_1 \cdot \delta(x_s = \text{Pidgey})x_{cp} \\ + b_2 \cdot \delta(x_s = \text{Weedle}) \\ + w_2 \cdot \delta(x_s = \text{Weedle})x_{cp} \\ + b_3 \cdot \delta(x_s = \text{Caterpie}) \\ + w_3 \cdot \delta(x_s = \text{Caterpie})x_{cp} \\ + b_4 \cdot \delta(x_s = \text{Eevee}) \\ + w_4 \cdot \delta(x_s = \text{Eevee})x_{cp}$$



其中  $\delta(x) = \begin{cases} 1, & \text{if } x \text{ 为 True} \\ 0, & \text{if } x \text{ 为 False} \end{cases}$

② 加入更多的影响因子进行建模：



## 6. Regularization

① 在尽量不增加模型复杂度的前提下，重新设计损失函数进行优化

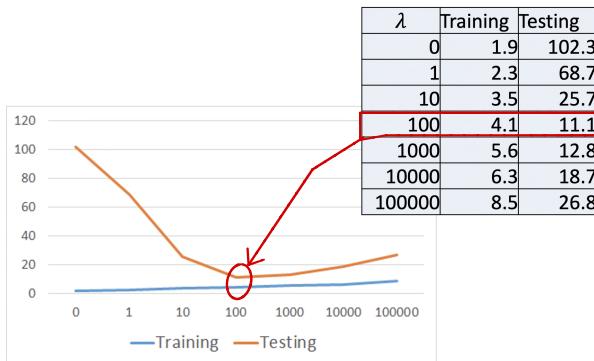
$$\text{模型: } y = b + \sum w_i x_i$$

$$\text{损失函数: } L = \sum_n \left( \hat{y}^n - (b + \sum w_i x_i) \right)^2 + \lambda \sum (w_i)^2$$

拥有更小  $w_i$  的模型，效果更好

对于模型而言，输入产生一个  $\Delta x_i$  的变化，则输出会产生  $w_i \Delta x_i$  的变化

更小的  $w_i$  → 函数更加平滑 → 对于输入中增加的干扰，越平滑的函数受到的影响越小



- $\lambda$  越大，更多考虑参数  $w$  本身，更少考虑  $x$
- Training error 放大  $\rightarrow$  Training Error  $\uparrow$   
↳ function 更平滑
- Testing Error  $\downarrow$

- 倾向于更平滑的函数，但不能过度平滑  
(由  $\lambda$  控制)

- bias 不需要加入正则项中，因为其不影响函数的平滑程度

## 7. Bias and Variance

① 假设数据本质上有一个某个未知函数  $f$ ，而我们通过训练数据得到的是  $f^*$ ，那么  $f^*$  是  $f$  的一定 estimator

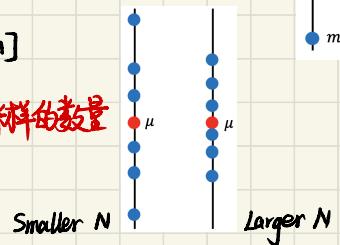
② 已知一个变量  $x$ ，其均值为  $\mu$ ，方差为  $\sigma^2$ ；选取  $x$  的  $N$  个样本： $\{x^1, x^2, x^3, \dots, x^N\}$

对均值  $\mu$  进行估计：

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

$m$  散布在  $\mu$  的周围，散布的紧密程度取决于  $\text{Var}[m]$

$$\text{Var}[m] = \frac{\sigma^2}{N}, \quad \text{Variance 取决于样本的数量}$$

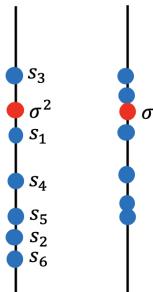


对方差 variance  $\sigma^2$  进行估计：

$$m = \frac{1}{N} \sum_{i=1}^n x_i^n \quad S = \frac{1}{N} \sum_{i=1}^n (x_i^n - m)^2$$

$$\text{有偏估计 } E[S] = \frac{N-1}{N} \cdot \sigma^2 \neq \sigma^2$$

Smaller N



Larger N

$$\begin{aligned} E[S^2] &= E\left[\frac{1}{n} \sum_{i=1}^n (X_i - \mu)^2 - \frac{2}{n}(\bar{X} - \mu) \sum_{i=1}^n (X_i - \mu) + (\bar{X} - \mu)^2\right] \\ &= E\left[\frac{1}{n} \sum_{i=1}^n (X_i - \mu)^2 - \frac{2}{n}(\bar{X} - \mu) \cdot n \cdot (\bar{X} - \mu) + (\bar{X} - \mu)^2\right] \\ &= E\left[\frac{1}{n} \sum_{i=1}^n (X_i - \mu)^2 - 2(\bar{X} - \mu)^2 + (\bar{X} - \mu)^2\right] \\ &= E\left[\frac{1}{n} \sum_{i=1}^n (X_i - \mu)^2\right] - E[(\bar{X} - \mu)^2] \\ &= E\left[\frac{1}{n} \sum_{i=1}^n (X_i - \mu)^2\right] - E[(\bar{X} - \mu)^2] \\ &= \sigma^2 - E[(\bar{X} - \mu)^2] \end{aligned}$$

补充：有偏估计与无偏估计

已知随机变量 X 的期望为  $\mu$ , 则方差  $\sigma^2$  为：

$$\sigma^2 = E[(X - \mu)^2]$$

因为 X 满足的具体分布并不清楚, 所以使用  $S^2$  估计  $\sigma^2$

$$S^2 = \frac{1}{n} \sum_{i=1}^n (X_i - \mu)^2$$

实际应用中, X 的期望  $\mu$  一般也是未知的, 只知道样本均值:

$$\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i, \quad E(\bar{X}) = \mu$$

因此可以这样计算  $S^2$ :

$$S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$$

使用样本均值  $\bar{X}$  代替 X 的期望,  $\frac{1}{n}$  变为  $\frac{1}{n-1}$  的原因为:

对于每次采样计算出的  $\bar{X}$  散布在  $\mu$  周围, 如果我们足够幸运, 某次采样使得  $\bar{X} = \mu$ , 则  $E[(X_i - \mu)^2]$  取最小值; 如果从偏高  $\bar{X}$ ,  $E[(X_i - \mu)^2]$  就会增大

其中,  $E[(\bar{X} - \mu)^2] = E[(\bar{X} - E(\bar{X}))^2]$ , 为样本均值的方差

假设样本  $X_1, X_2, \dots, X_n$  独立同分布, 每个样本的方差相等, 即

$$\text{Var}(X_1) = \text{Var}(X_2) = \dots = \text{Var}(X_n)$$

$$\text{于是, } \text{Var}(\bar{X}) = \text{Var}\left(\frac{1}{n}(X_1 + X_2 + \dots + X_n)\right) = \frac{\text{Var}(X_1 + X_2 + \dots + X_n)}{n} = \frac{1}{n^2} [\text{Var}(X_1) + \text{Var}(X_2) + \dots + \text{Var}(X_n)] = \frac{1}{n^2} \cdot n \text{Var}(X_1) = \frac{1}{n} \text{Var}(X_1) = \frac{1}{n} \sigma^2$$

$\bar{X}$  服从于  $(\mu, \frac{\sigma^2}{n})$  的正态分布

$$\text{所以 } E\left[\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2\right] = \sigma^2 - \frac{1}{n} \sigma^2 = \frac{n-1}{n} \sigma^2$$

由于低估了  $\frac{1}{n} \sigma^2$ , 进行调整:

$$\frac{n}{n-1} E\left[\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2\right] = E\left[\frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2\right] = \sigma^2$$

所以  $S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$  就是方差的无偏估计

$$\text{所以 } \sum_{i=1}^n (X_i - \bar{X})^2 \leq \sum_{i=1}^n (X_i - \mu)^2$$

$$\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2 \leq \frac{1}{n} \sum_{i=1}^n (X_i - \mu)^2$$

$$E\left[\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2\right] \leq E\left[\frac{1}{n} \sum_{i=1}^n (X_i - \mu)^2\right] = \sigma^2$$

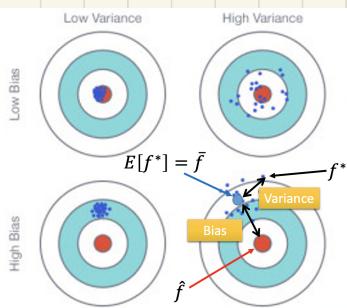
由此可见:  $S^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X})^2$  倾向于低估  $\sigma^2$ , 是有偏的

# 使用 $\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2$ 对 $\sigma^2$ 进行估计的权重

$$\begin{aligned}\mathbb{E}[S^2] &= \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2\right] = \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^n ((X_i - \mu) - (\bar{X} - \mu))^2\right] \\ &= \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^n ((X_i - \mu)^2 - 2(\bar{X} - \mu)(X_i - \mu) + (\bar{X} - \mu)^2)\right] \\ &= \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^n (X_i - \mu)^2 - \frac{2}{n}(\bar{X} - \mu) \sum_{i=1}^n (X_i - \mu) + \frac{1}{n}(\bar{X} - \mu)^2 \sum_{i=1}^n 1\right] \\ &= \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^n (X_i - \mu)^2 - \frac{2}{n}(\bar{X} - \mu) \sum_{i=1}^n (X_i - \mu) + \frac{1}{n}(\bar{X} - \mu)^2 \cdot n\right] \\ &= \mathbb{E}\left[\frac{1}{n} \sum_{i=1}^n (X_i - \mu)^2 - \frac{2}{n}(\bar{X} - \mu) \sum_{i=1}^n (X_i - \mu) + (\bar{X} - \mu)^2\right]\end{aligned}$$

其中,  $\bar{X} - \mu = \frac{1}{n} \sum_{i=1}^n X_i - \mu = \frac{1}{n} \sum_{i=1}^n X_i - \frac{1}{n} \sum_{i=1}^n \mu = \frac{1}{n} \sum_{i=1}^n (X_i - \mu)$

## ③ 偏差与方差的作用:



$\hat{f}$ : 变量的真实分布, 不可知

$f^*$ : 通过训练找到的模型函数

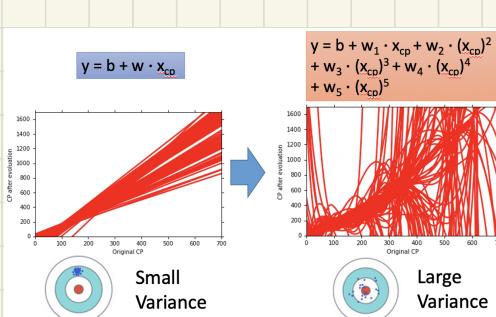
$\bar{f}: f^*$  的期望

$\bar{f}$  与  $\hat{f}$  之间的差距是由 Bias 决定的, 是系统性的偏差, 例如你的准星不准, 即使用厉害也打不中红心

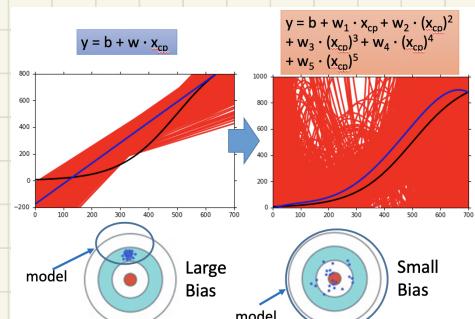
$f^*$  与  $\bar{f}$  之间的差距是由 Variance 决定的, 例如风速, 阻力等

最理想的方案: Low Bias & Low Variance

每次找 10 只金鱼进行预测, 进行 100 次实验可以得到不同模型的差异:



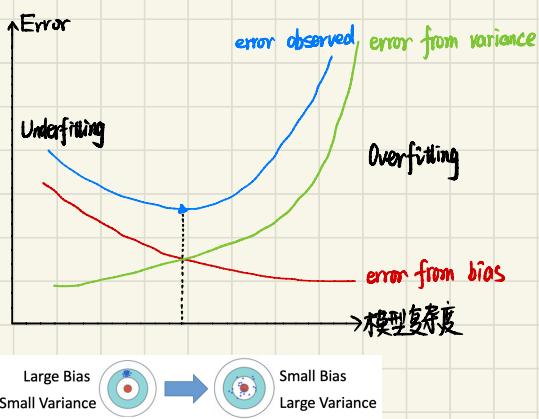
越简单的模型, 受到不同抽样数据的影响越小



黑色:  $\hat{f}$  蓝色:  $\bar{f}: E[\hat{f}]$  红色: 500个  $\hat{f}^*$

越简单的模型, bias越大, 模型的空间越小, 不论怎么找也不会命中红心

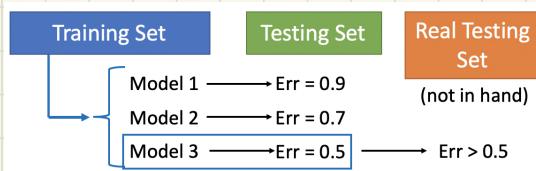
越复杂的模型, bias越小, 模型的空间越大, 有可能命中红心



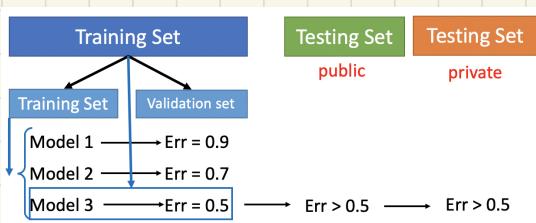
- 如果模型在训练样本上都无法很好的拟合，有可能是欠拟合，需要重新设计更复杂的模型
- 如果训练误差小，测试误差大，有可能是过拟合，需要更多的训练数据（很有效的方法，但收集更多的数据比较困难；或是人工生成更多 training data，旋转图像/加噪音）或是加入正则项（Regularization，调整正则项的权重在 bias 和 variance 中取得平衡，图中的 ▲）

## 8. Model Selection

### ① 交叉验证 (Cross Validation)



• 在一个测试集上取得较好的结果，不代表在真实数据上效果也很好



- 将 Training Set 分为两部分，一部分训练，一部分验证。选择模型后，再使用全部 Training Set 重新训练一次，然后跑 Testing Set
- 即使在 public Testing Set 上效果不好，仍不建议对模型参数进行一些修改降低该部分的误差。因为这仍然是修正了 public Testing set 中数据的 noise，对于 private Testing Set 并不起作用

### ② N-fold Cross Validation: 由于拆分过程可能导致 Validation Set 中有一些影响模型的 noise，所以可以采用 k 倍交叉验证

| Training Set       | Model 1       |       |     | Model 2             |     |       | Model 3       |     |       |
|--------------------|---------------|-------|-----|---------------------|-----|-------|---------------|-----|-------|
|                    | Train         | Train | Val | Train               | Val | Train | Train         | Val | Train |
|                    | Err = 0.2     |       |     | Err = 0.4           |     |       | Err = 0.4     |     |       |
|                    | Err = 0.4     |       |     | Err = 0.5           |     |       | Err = 0.5     |     |       |
|                    | Err = 0.3     |       |     | Err = 0.6           |     |       | Err = 0.3     |     |       |
|                    | Avg Err = 0.3 |       |     | Avg Err = 0.5       |     |       | Avg Err = 0.4 |     |       |
| Testing Set public |               |       |     | Testing Set private |     |       |               |     |       |

## 9. Homework 1

### ① 数据标准化

归一化：Min-Max标准化 / 0-1标准化 / 差值标准化（通过线性变换使结果落在[0,1]之间）

$$x' = \frac{x - \min\{x\}}{\max\{x\} - \min\{x\}}$$

- $x'$  无量纲
- 有新数据加入时可能影响 min/max，需重新计算

标准化：Z-Score 标准化

$$x' = \frac{x - \bar{x}}{s}$$

- $\bar{x}$  为原数据的均值， $s$  为原数据的标准差；处理后的数据服从(0,1)正态分布
- Z-Score 标准化适用于数据的 min 或 max 未知的情况，或有较大取值范围的数据点
- Z-Score 标准化要求原数据近似服从高斯分布

### ② 损差的种类

#### RMSE (Root Mean Square Error) 均方根误差

衡量观测值与真实值之间的偏差。

常用来作为机器学习模型预测结果衡量的标准。

$$RMSE(X, h) = \sqrt{\frac{1}{m} \sum_{i=1}^m (h(x_i) - y_i)^2}$$

#### MSE (Mean Square Error) 均方误差

MSE 是真实值与预测值的差值的平方然后求和平均。

通过平方的形式便于求导，所以常被用作线性回归的损失函数

$$MSE = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2$$

#### MAE (Mean Absolute Error) 平均绝对误差

是绝对误差的平均值。

可以更好地反映预测值误差的实际情况。

$$MAE(X, h) = \frac{1}{m} \sum_{i=1}^m |h(x_i) - y_i|$$

#### SD (Standard Deviation) 标准差

方差的算术平均根。

用于衡量一组数值的离散程度。

$$SD = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - avg(x))^2}$$

### ③ 梯度下降法优化方法：

深度学习常见的优化方法(Optimizer)总结:Adam,SGD,Momentum,AdaGard等：

<https://www.cnblogs.com/GeekDanny/p/9655597.html>

sigmoid函数的输入为 z:  $\text{sig}(z)$ , 损失函数  $f(w) = \frac{1}{2} \sum_{i=1}^m (z(x_i) - y_i)^2$ , 因为用梯度上升, 故  $J(w) = -f(w)$

$$z = w_0 + w_1 x_1 + w_2 x_2 + \dots + w_n x_n \quad (n \text{ 个特征})$$

$$z = w^T x = [w_0 \ w_1 \ \dots \ w_n] \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$$

$$= x^T w = [x_0 \ x_1 \ \dots \ x_n] \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_n \end{bmatrix}$$

$$\text{令 } X = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_m^T \end{bmatrix}, \text{ 则 } x^T w = \begin{bmatrix} x_1^T \\ x_2^T \\ \vdots \\ x_m^T \end{bmatrix} w = \begin{bmatrix} x_1^T w \\ x_2^T w \\ \vdots \\ x_m^T w \end{bmatrix} = \begin{bmatrix} z(x_1) \\ z(x_2) \\ \vdots \\ z(x_m) \end{bmatrix}$$

令  $\vec{y}$  表示样本的实际标签:

$$x^T w - \vec{y} = \begin{bmatrix} z(x_1) - y_1 \\ z(x_2) - y_2 \\ \vdots \\ z(x_m) - y_m \end{bmatrix}$$

$$\because \alpha^T \alpha = \sum_i^n \alpha_i^2 \quad \therefore \frac{1}{2} (x^T w - \vec{y})^T (x^T w - \vec{y}) = \frac{1}{2} \sum_{i=1}^m (z(x_i) - y_i)^2 = f(w)$$

$$\text{则梯度: } \nabla_w f(w) = \nabla_w \left[ \frac{1}{2} (x^T w - \vec{y})^T (x^T w - \vec{y}) \right]$$

$$= \frac{1}{2} \nabla_w \left[ \underbrace{w^T x^T x w}_{1 \times 1 \text{ 矩阵}} - \underbrace{\frac{1}{2} w^T x^T \vec{y}}_{1 \times 1 \text{ 矩阵}} - \underbrace{\vec{y}^T x w}_{1 \times 1} + \underbrace{\frac{1}{2} \vec{y}^T \vec{y}}_{1 \times 1} \right]$$

$\because$  括号内结果为  $1 \times 1$  方阵, 其值等于矩阵的迹, 所以上式可变换为:

$$= \frac{1}{2} \nabla_w [w^T x^T x w - w^T x^T \vec{y} - \vec{y}^T x w + \underbrace{\vec{y}^T \vec{y}}_{\cancel{\text{与 } w \text{ 无关}}}].$$

$$= \frac{1}{2} \nabla_w \left[ w^T x^T x w - w^T x^T \vec{y} - \vec{y}^T x w \right]$$

$$\because \text{tr } A = \text{tr } A^T \quad \therefore \text{tr } (w^T x^T \vec{y}) = \text{tr } (\vec{y}^T x w)$$

$$\because \text{tr } (A + B) = \text{tr } A + \text{tr } B$$

故上式为:

$$= \frac{1}{2} \nabla_w (\text{tr } w^T x^T x w - 2 \text{tr } \vec{y}^T x w)$$

$$\because \text{tr } (ABC) = \text{tr } (CAB) = \text{tr } (BCA)$$

$$\therefore \text{tr } \vec{y}^T x w = \text{tr } w \vec{y}^T x$$

$$\therefore \text{tr } (AB) = B^T \quad \therefore \text{tr } w \vec{y}^T x = (x^T \vec{y})$$

故上式为:

$$= \frac{1}{2} \nabla_w (\text{tr } w^T x^T x w - (x^T \vec{y}))$$

$$\therefore \nabla_A \text{tr } (ABATC) = CAB + C^T A B^T$$

$$\text{或: } \frac{\partial x^T A x}{\partial x} = (A + A^T) x$$

$$\therefore \nabla_w (\text{tr } w^T x^T x w) = [(x^T x + (x^T x)^T) w] = 2x^T x w$$

$$\text{故上式: } = \cancel{x^T x w} - x^T \vec{y}$$

$$= x^T (x^T w - \vec{y})$$