Regression_leeHungyi

2020年7月22日 17:45

Regression: Output a scalar

- Stock Market Forecast 股票预测

$f($  $) =$ Dow Jones Industrial Average at tomorrow

明天 明天道琼斯工业平均指数

→ 过去10年… 的资料

- Self-driving Car 无人车 / 自动车

$f($  $) =$ 方向盤角度 → 如何行驶

无人车感受到的信息

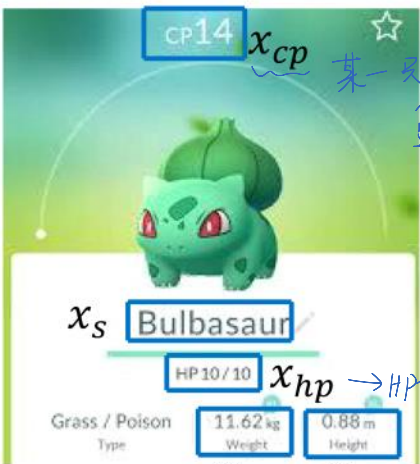
- Recommendation 推荐系统

$f($ 使用者 A 商品 B $) =$ 購買可能性

Example Application

预测 CP值

- Estimating the Combat Power (CP) of a pokemon after evolution

$f($ 

$) =$ CP after evolution y

x_{cp} 某一只之间的CP值 进化前

x_s

x_{hp} → HP值

x_w 重量

x_h 高度

x 某一只宝可梦

进化后的CP值

Step 1: Model

w and b are parameters
(can be any value)

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

进化后的cp值 进化前的cp值

A set of function

Model

$f_1, f_2 \cdots$

f_1: $y = 10.0 + 9.0 \cdot x_{cp}$

f_2: $y = 9.8 + 9.2 \cdot x_{cp}$

f_3: $y = -0.8 - 1.2 \cdot x_{cp}$

...... infinite

$f($ 

$x) =$ CP after evolution y

属性

an attribute of input x

x_i: $x_{cp}, x_{hp}, x_w, x_h \ldots$

feature

Linear model: $y = b + \sum w_i x_i$

线性模型

w_i: weight, b: bias

Step 2: Goodness of Function

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

A set of function

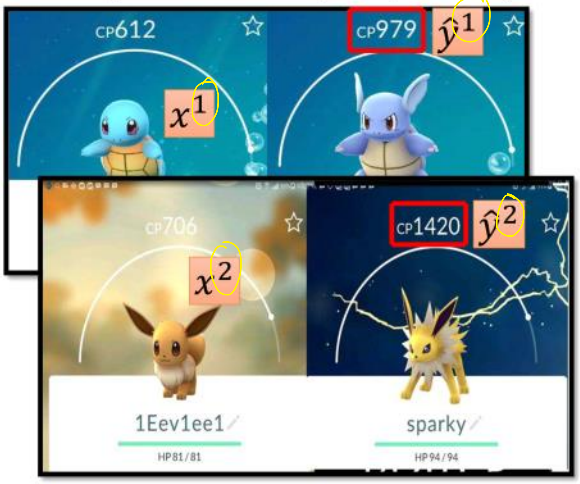
Model

$f_1, f_2 \cdots$

0 不同个体

function input:

function Output (scalar):



Training Data

Step 2: Goodness of Function

Training Data:
10 pokemons

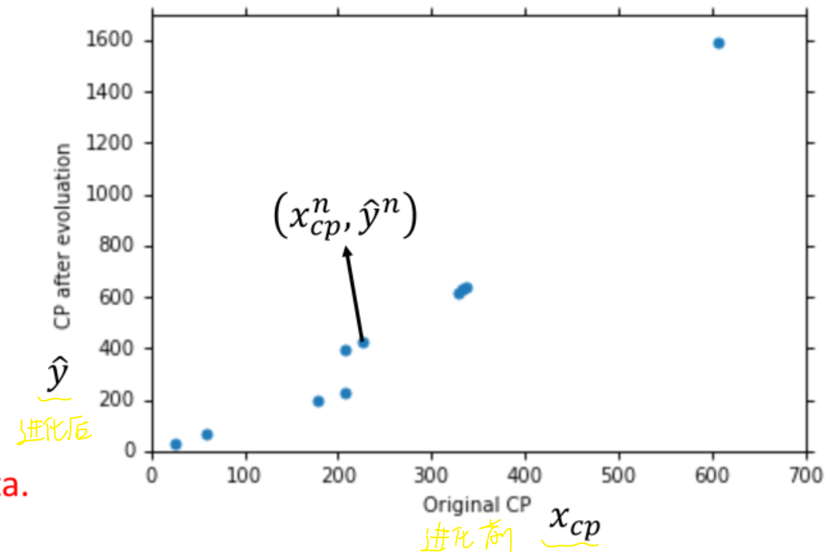
$(x^1, \hat{y}^1)$

$(x^2, \hat{y}^2)$

$\vdots$

$(x^{10}, \hat{y}^{10})$

This is real data.



Source: https://www.openintro.org/stat/data/?data=pokemon

Step 2: Goodness of Function

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

A set of function → Model: $f_1, f_2 \cdots$

↓

Goodness of function f

↑

Training Data

Loss function L:

Input: a function, output: how bad it is

$L(f) = L(w, b)$

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

Estimation error 估测误差

$$L(f) = \sum_{n=1}^{10} \boxed{\left(\hat{y}^n - f(x_{cp}^n)\right)^2}$$

Sum over examples

Estimated y based on input function

$$L(w, b) = \sum_{n=1}^{10} \left(\underbrace{\hat{y}^n}_{\text{真的}} - \underbrace{(b + w \cdot x_{cp}^n)}_{\text{预测的}}\right)^2$$

Step 2: Goodness of Function

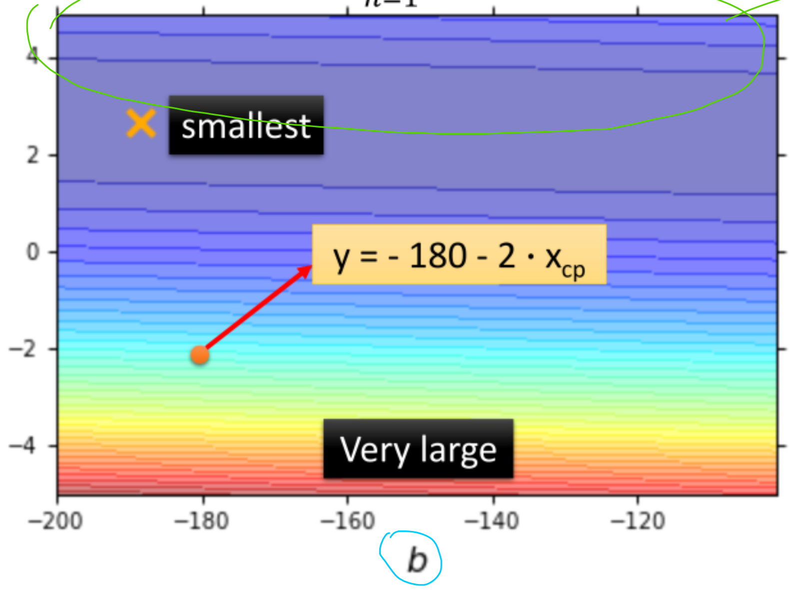
- Loss Function

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

Each point in the figure is a function

The color represents $L(w,b)$.

(true example)



最好的function

小

每个点都是个 L(w,b)

大

Step 3: Best Function

A set of function

Model

$f_1, f_2 \cdots$

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

哪个 w . b 使 L(w,b) 最小

Goodness of function f

Pick the "Best" Function

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

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

$$= arg \min_{w,b} \sum_{n=1}^{10} \left(\hat{y}^n - (b + w \cdot x_{cp}^n)\right)^2$$

Training Data

梯度下降 →

Gradient Descent

只要是可微分的，都可用

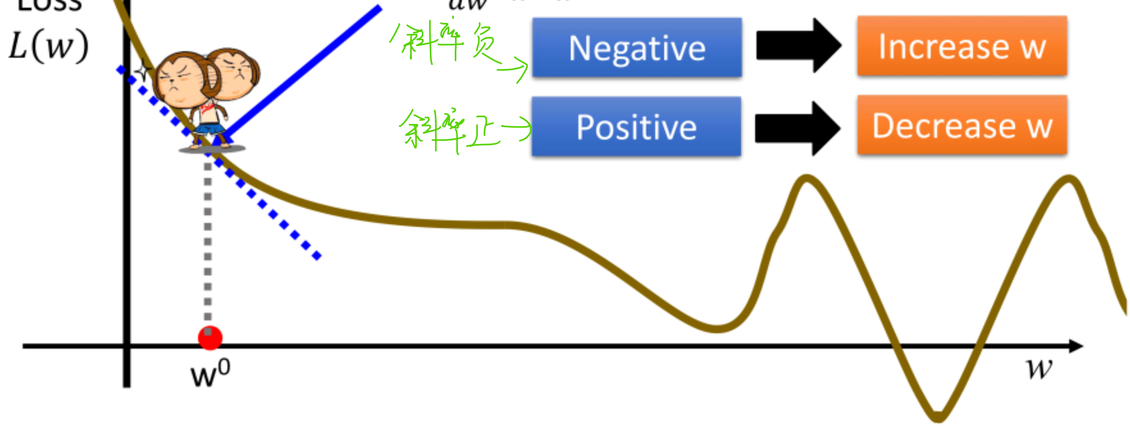
Step 3: Gradient Descent

$$w^* = arg \min_w L(w)$$

- Consider loss function $L(w)$ with one parameter w:
  - ➢ (Randomly) Pick an initial value w^0 随机初始化 W_0
  - ➢ Compute $\frac{dL}{dw}|_{w=w^0}$ 计算 W_0 处的偏导数/微分 → 也就是切线斜率

斜率负 → Negative ➡ Increase w

斜率正 → Positive ➡ Decrease w



http://chico386.pixnet.net/album/photo/171572850

Step 3: Gradient Descent

$$w^* = arg \min_w L(w)$$

- Consider loss function $L(w)$ with one parameter w:
  - ➢ (Randomly) Pick an initial value w^0
  - ➢ Compute $\frac{dL}{dw}|_{w=w^0}$

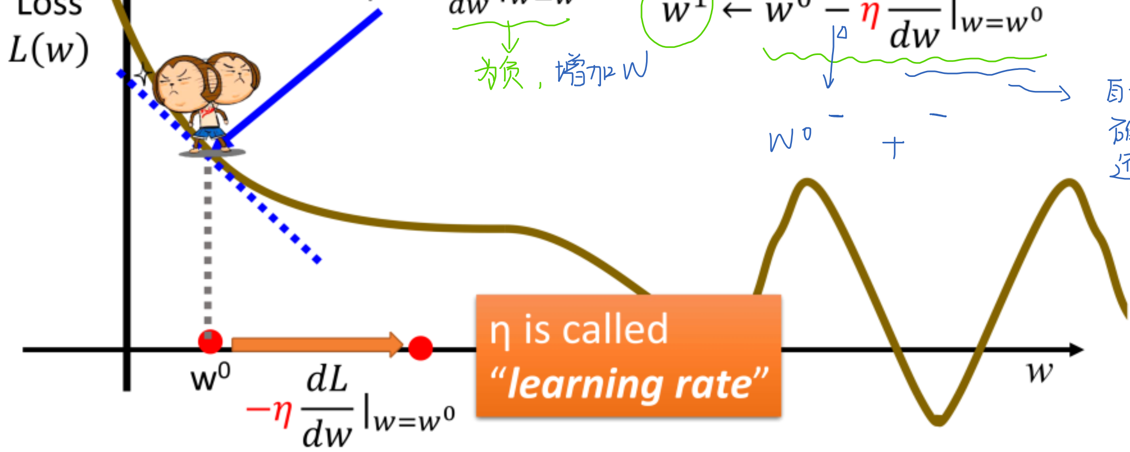
learning rate

$$w^1 \leftarrow w^0 - \eta \frac{dL}{dw}|_{w=w^0}$$

为负，增加 w

w^0 − +

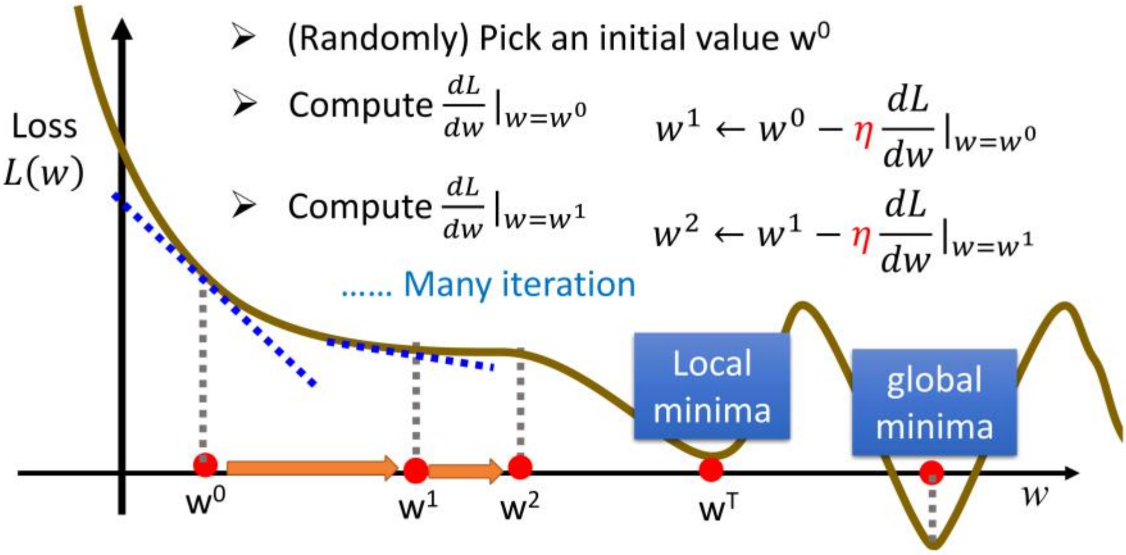
自己带有符号 确定w是增加还是减少



Step 3: Gradient Descent

$$w^* = arg \min_w L(w)$$

- Consider loss function $L(w)$ with one parameter w:



> (Randomly) Pick an initial value w^0

> Compute $\frac{dL}{dw}|_{w=w^0}$ $\quad w^1 \leftarrow w^0 - \eta \frac{dL}{dw}|_{w=w^0}$

> Compute $\frac{dL}{dw}|_{w=w^1}$ $\quad w^2 \leftarrow w^1 - \eta \frac{dL}{dw}|_{w=w^1}$

...... Many iteration

Step 3: Gradient Descent

$$\nabla L = \begin{bmatrix} \frac{\partial L}{\partial w} \\ \frac{\partial L}{\partial b} \end{bmatrix} \text{gradient}$$

- How about two parameters? $\quad w^*, b^* = arg \min_{w,b} L(w, b)$

> (Randomly) Pick an initial value w^0, b^0

> Compute $\frac{\partial L}{\partial w}|_{w=w^0,b=b^0}$, $\frac{\partial L}{\partial b}|_{w=w^0,b=b^0}$

$$w^1 \leftarrow w^0 - \eta \frac{\partial L}{\partial w}|_{w=w^0,b=b^0} \qquad b^1 \leftarrow b^0 - \eta \frac{\partial L}{\partial b}|_{w=w^0,b=b^0}$$

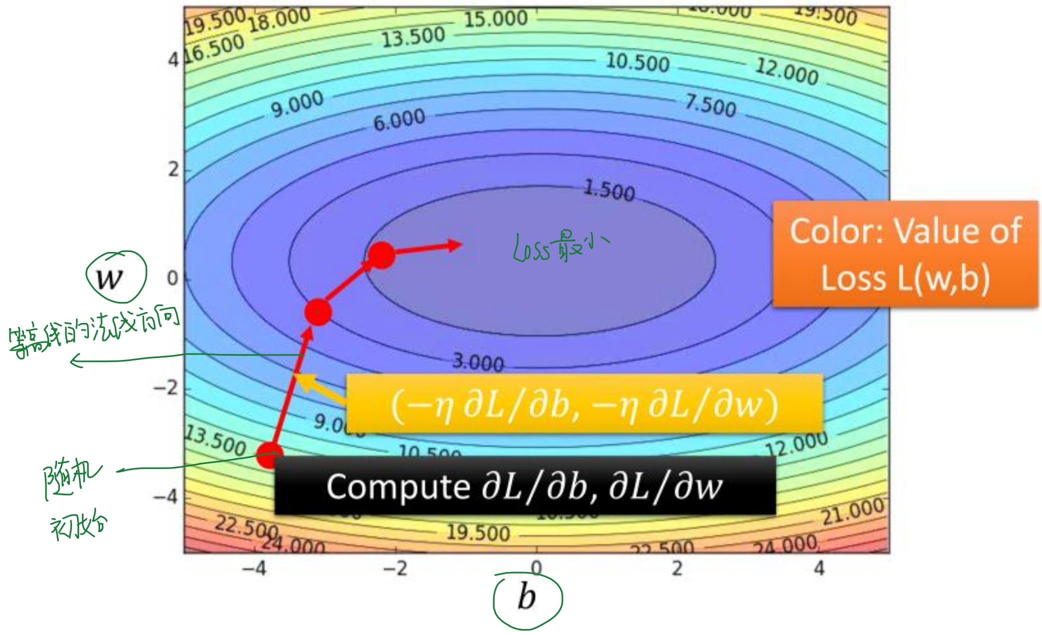
更新w^1 b^1

> Compute $\frac{\partial L}{\partial w}|_{w=w^1,b=b^1}$, $\frac{\partial L}{\partial b}|_{w=w^1,b=b^1}$

$$w^2 \leftarrow w^1 - \eta \frac{\partial L}{\partial w}|_{w=w^1,b=b^1} \qquad b^2 \leftarrow b^1 - \eta \frac{\partial L}{\partial b}|_{w=w^1,b=b^1}$$

更新w^2 b^2

Step 3: Gradient Descent



Step 3: Gradient Descent

- When solving:

$$\theta^* = \arg\max_{\theta} L(\theta) \quad \text{by gradient descent}$$

- Each time we update the parameters, we obtain θ that makes $L(\theta)$ smaller.

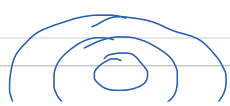
$$L(\theta^0) > L(\theta^1) > L(\theta^2) > \cdots$$

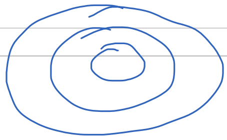
Is this statement correct?

In linear regression the loss function is Convex → 无 local optimal

线性回归 凸函数 无局部最优解

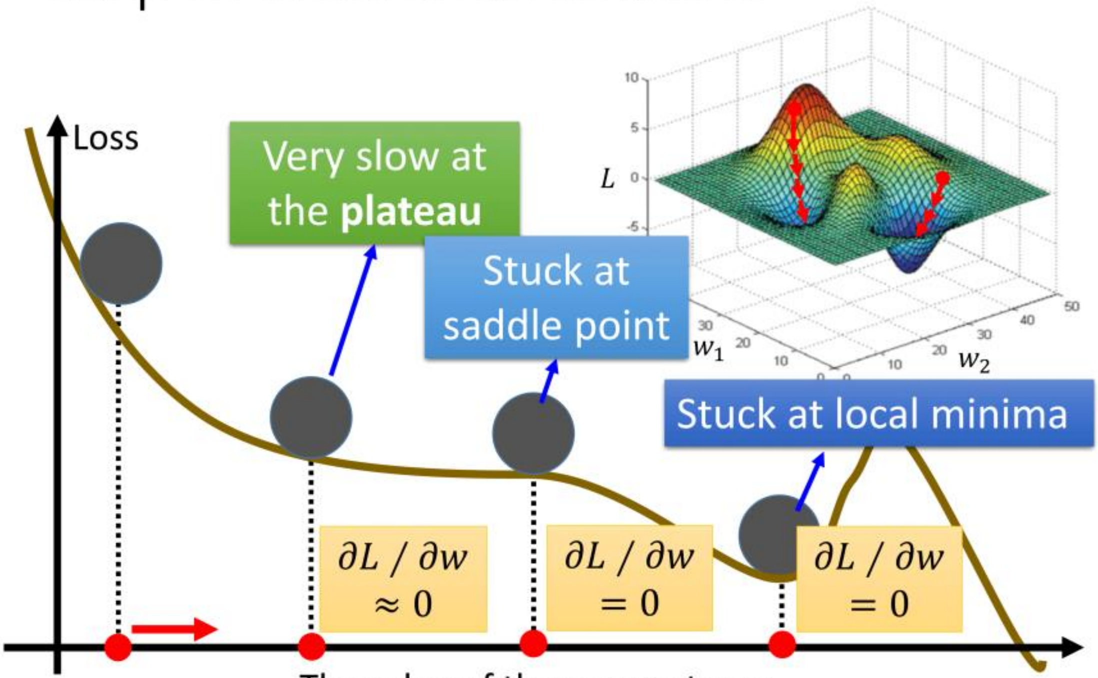
等高线是一圈





等高线是一圈一圈的

Step 3: Gradient Descent



Step 3: Gradient Descent

- Formulation of $\partial L/\partial w$ and $\partial L/\partial b$

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

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

$$\frac{\partial L}{\partial w} = \sum 2\left(\hat{y}^n - (b + w \cdot x_{cp}^n)\right) \cdot \left(-x_{cp}^n\right)$$

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

Step 3: Gradient Descent

- Formulation of $\partial L / \partial w$ and $\partial L / \partial b$

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

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

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

Step 3: Gradient Descent

How's the results?

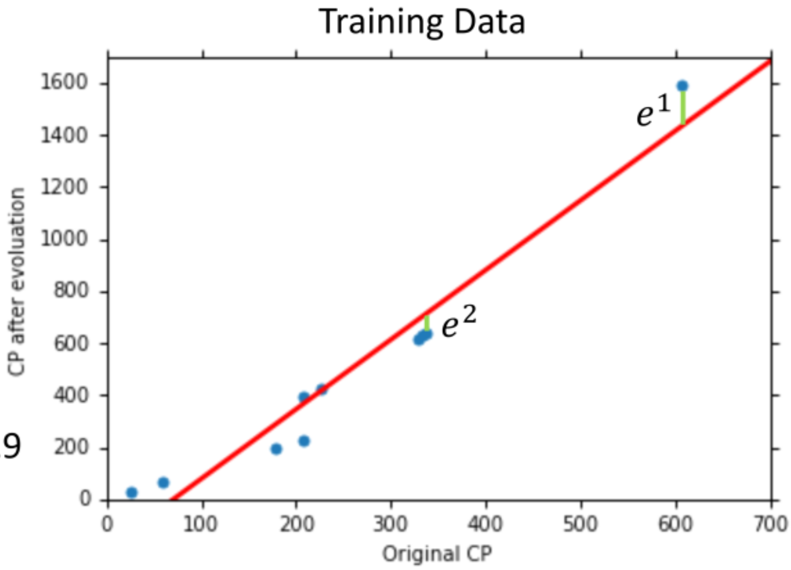
$y = b + w \cdot x_{cp}$

b = -188.4

w = 2.7

Average Error on Training Data

$$= \frac{1}{10}\sum_{n=1}^{10} e^n = 31.9$$



How's the results? - Generalization

What we really care about is the error on new data (testing data)

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

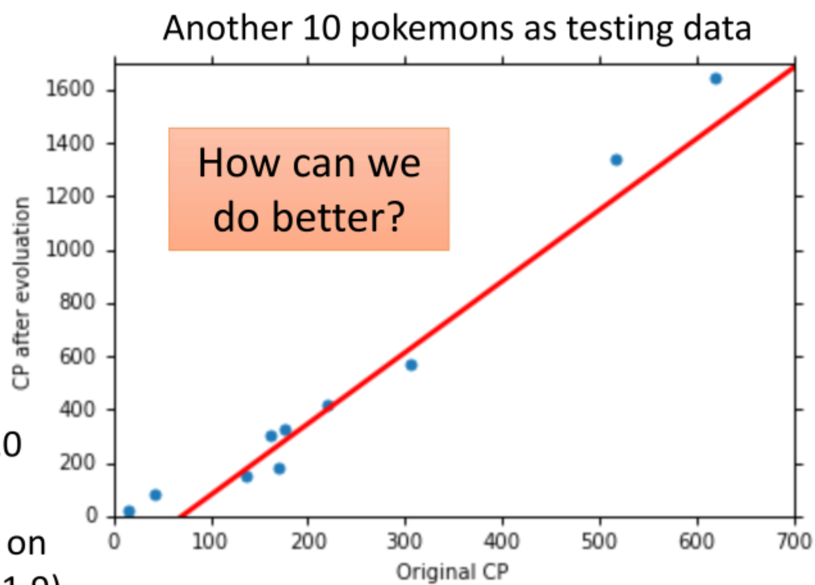
b = -188.4

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

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

Best Function

$b = -10.3$

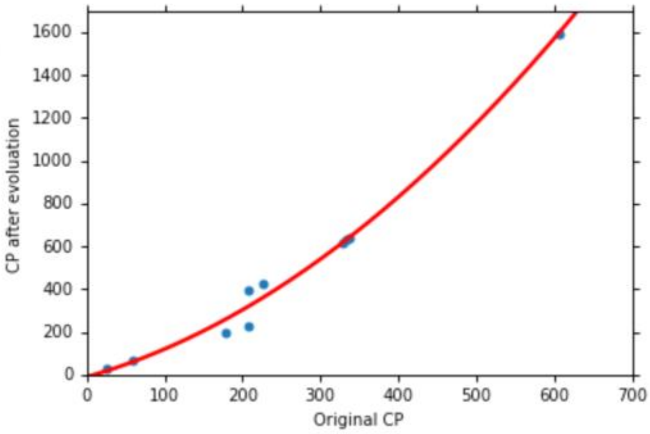
$w_1 = 1.0$, $w_2 = 2.7 \times 10^{-3}$

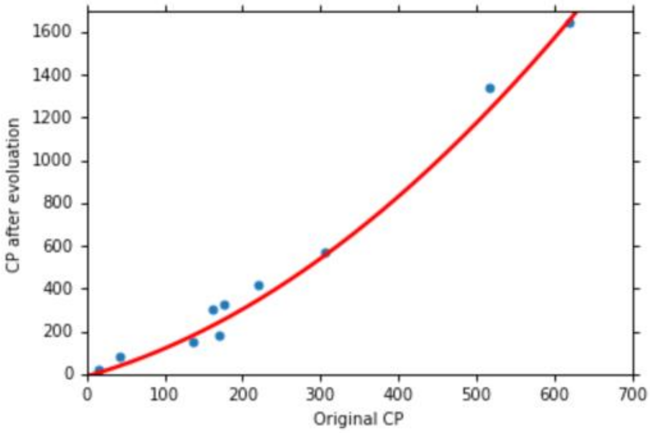
Average Error = 15.4

Testing:

Average Error = 18.4

Better! Could it be even better?





Selecting another Model

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

Best Function

$b = 6.4$, $w_1 = 0.66$

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

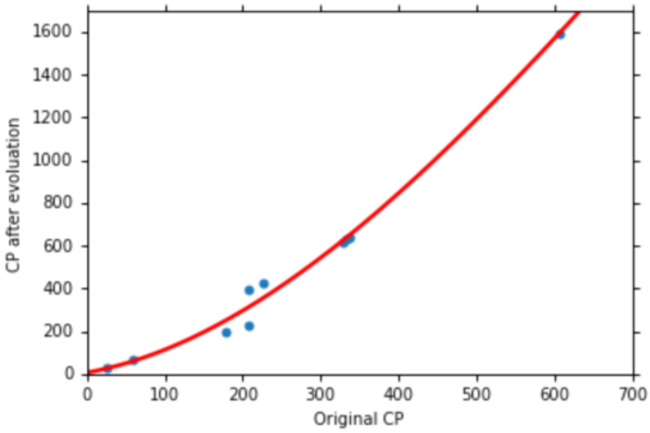
$w_3 = -1.8 \times 10^{-6}$

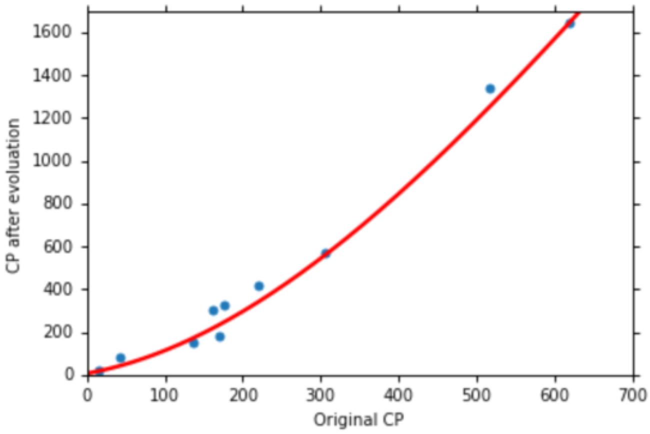
Average Error = 15.3

Testing:

Average Error = 18.1

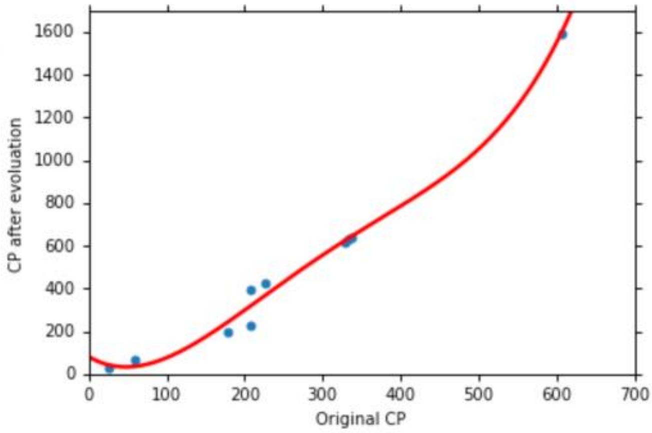
Slightly better.
How about more complex model?





Selecting another Model

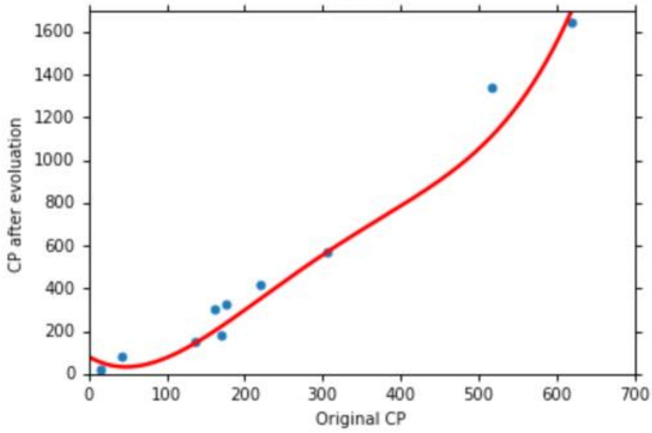
$$y = b + w_1 \cdot x_{cp} + w_2 \cdot (x_{cp})^2 + w_3 \cdot (x_{cp})^3 + w_4 \cdot (x_{cp})^4$$



Best Function

Average Error = 14.9

Testing:

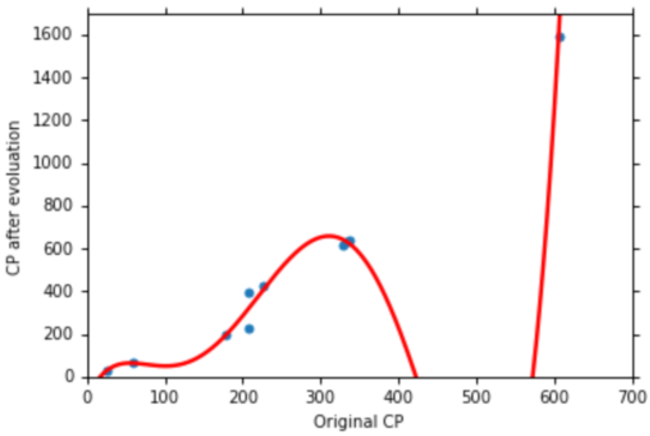


Average Error = 28.8

The results become worse ...

Selecting another Model

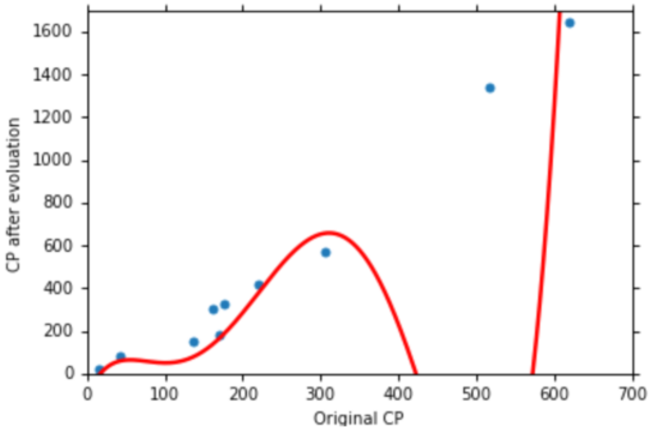
$$y = b + w_1 \cdot x_{cp} + w_2 \cdot (x_{cp})^2 + w_3 \cdot (x_{cp})^3 + w_4 \cdot (x_{cp})^4 + w_5 \cdot (x_{cp})^5$$



Best Function

Average Error = 12.8

Testing:



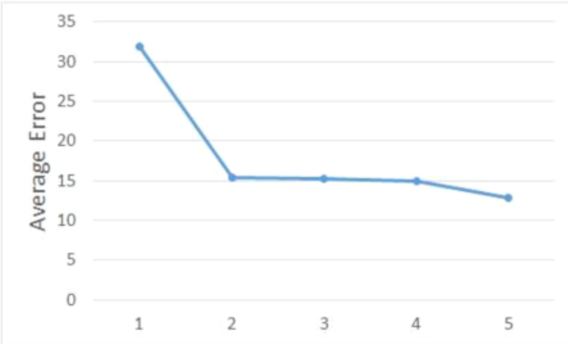
Average Error = 232.1

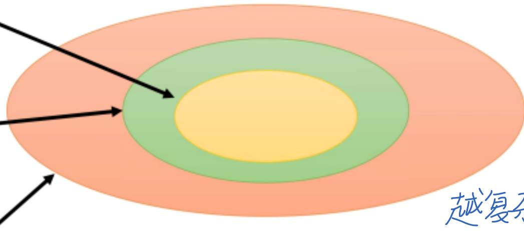
The results are so bad.

Model Selection

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

Training Data



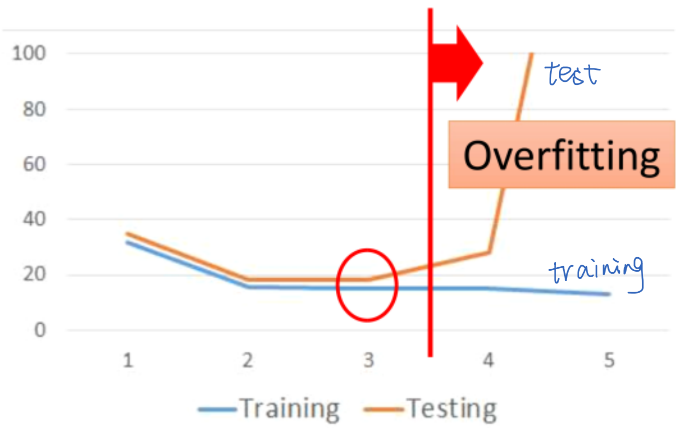


越复杂的model ↓

A more complex model yields lower error on training data. function越复杂

If we can truly find the best function

Model Selection



	Training	Testing
1	31.9	35.0
2	15.4	18.4
3	15.3	18.1
4	14.9	28.2
5	12.8	232.1

model

并不

A more complex model does not always lead to better performance on ***testing data***.

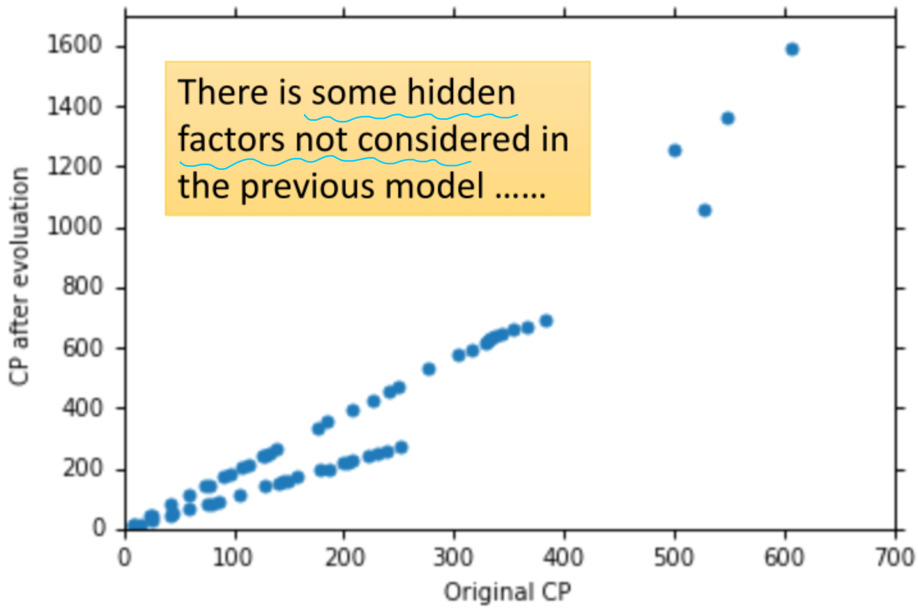
This is ***Overfitting***. → Select suitable model → 合适的model

过拟合 → 模型越复杂，但error不是最优

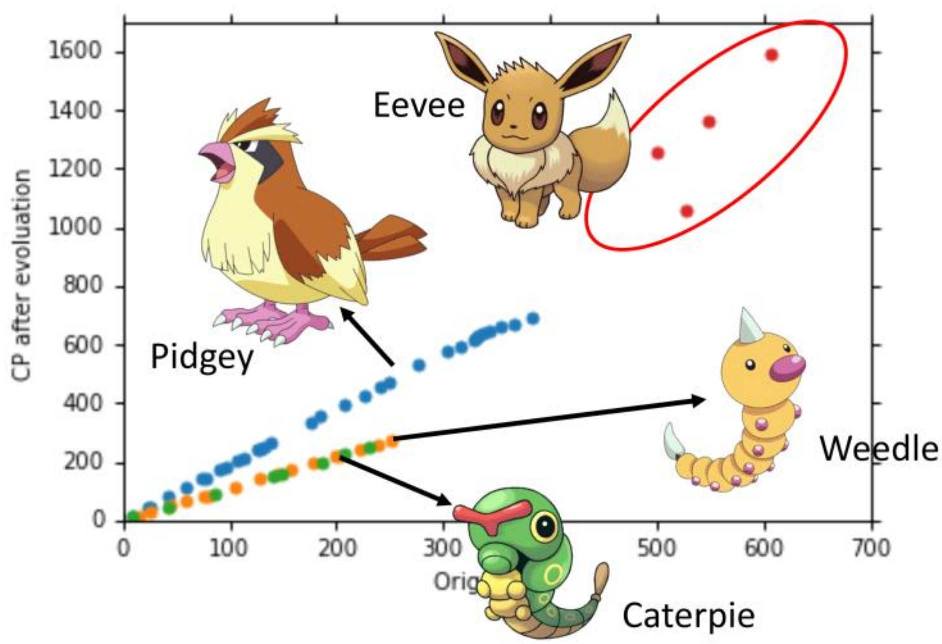
test的error很少
但training的error较大

、test的error很少
但training的error较大

Let’s collect more data



What are the hidden factors?



Back to step 1: Redesign the Model

重新设计 function

$$y = b + \sum w_i x_i$$

Linear model?

x_s = species of x 4个物种

x

If x_s = Pidgey: $y = b_1 + w_1 \cdot x_{cp}$

If x_s = Weedle: $y = b_2 + w_2 \cdot x_{cp}$

If x_s = Caterpie: $y = b_3 + w_3 \cdot x_{cp}$

If x_s = Eevee: $y = b_4 + w_4 \cdot x_{cp}$

不同的物种有不同的y

y

Back to step 1: Redesign the Model

$$y = b + \sum w_i x_i$$

Linear model?

$b_1 \quad b_2 \quad b_3 \quad b_4$

$w_1 \quad w_2 \quad w_3 \quad w_4$

$y = b_1 \cdot \delta(x_s = \text{Pidgey})$

$+ w_1 \cdot \delta(x_s = \text{Pidgey}) x_{cp}$

$+ b_2 \cdot \delta(x_s = \text{Weedle})$

$+ w_2 \cdot \delta(x_s = \text{Weedle}) x_{cp}$

$+ b_3 \cdot \delta(x_s = \text{Caterpie})$

$+ w_3 \cdot \delta(x_s = \text{Caterpie}) x_{cp}$

$+ b_4 \cdot \delta(x_s = \text{Eevee})$

$+ w_4 \cdot \delta(x_s = \text{Eevee}) x_{cp}$

$$\delta(x_s = \text{Pidgey}) \begin{cases} =1 \quad 1 & \text{If } x_s = \text{Pidgey} \\ =0 \quad 0 & \text{otherwise} \end{cases}$$

Back to step 1: Redesign the Model

$$y = b + \sum w_i \boxed{x_i}$$

Linear model?

$y = b_1 \cdot$ 1

$+w_1 \cdot$ 1 x_{cp}

$+b_2 \cdot$ 0

$+w_2 \cdot$ 0

$+b_3 \cdot$ 0

$+w_3 \cdot$ 0

$+b_4 \cdot$ 0

$+w_4 \cdot$ 0

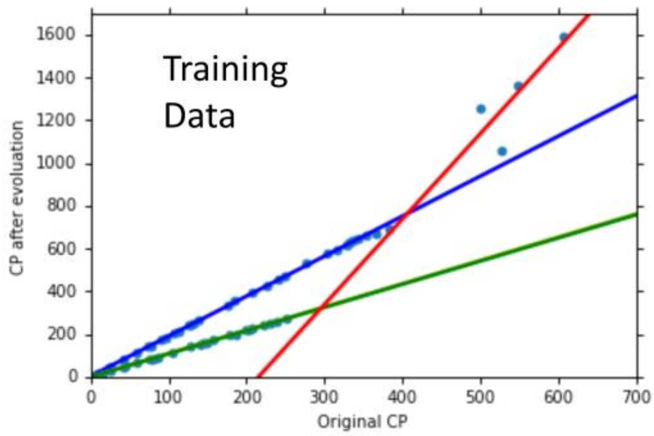
$\delta(x_s = \text{Pidgey})$

$$\begin{cases} =1 & \text{If } x_s = \text{Pidgey} \\ =0 & \text{otherwise} \end{cases}$$

If $x_s = \text{Pidgey}$

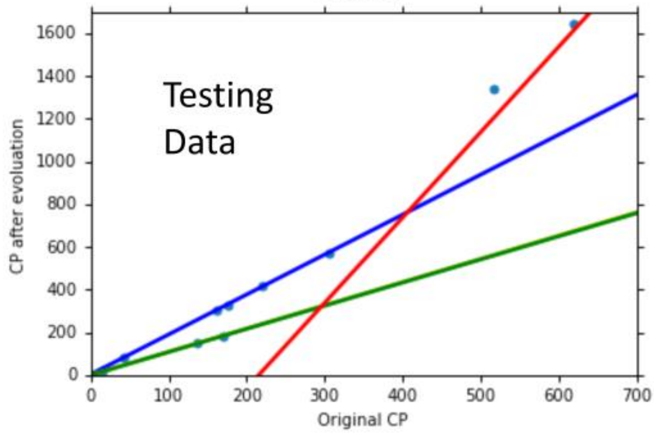
$$y = b_1 + w_1 \cdot x_{cp}$$

Average error = 3.8



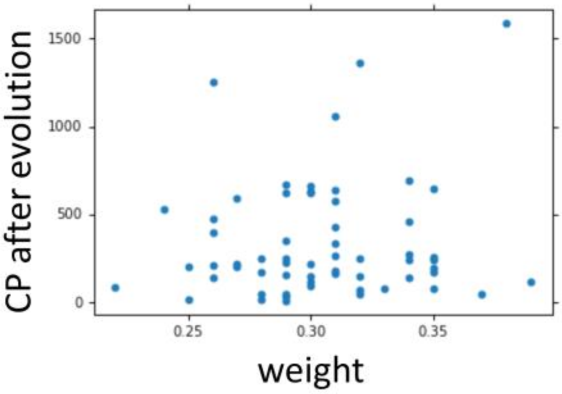
不同种类
有不同的model

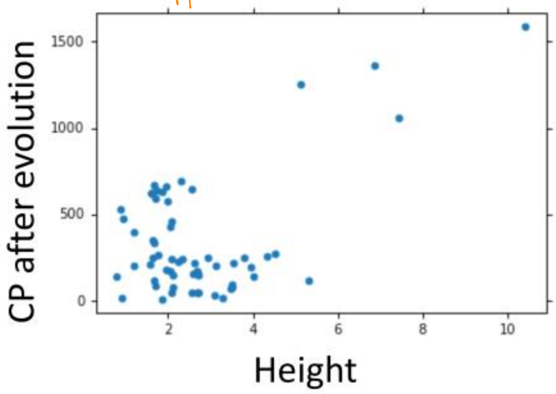
Average error = 14.3

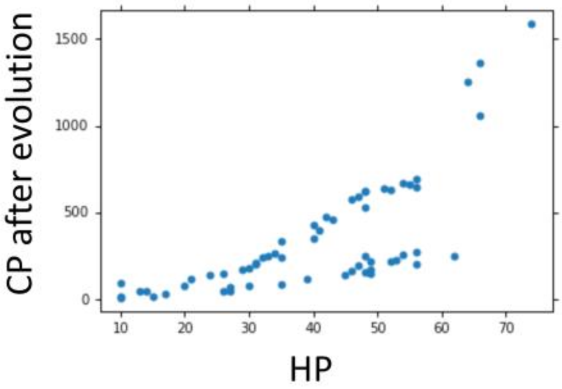


Are there any other hidden factors?

$y \sim$ weight, height, HP 是否有关







Back to step 1: Redesign the Model Again

将模型变复杂一点

x

不是y

If x_s = Pidgey: $y' = b_1 + w_1 \cdot x_{cp} + w_5 \cdot (x_{cp})^2$

If x_s = Weedle: $y' = b_2 + w_2 \cdot x_{cp} + w_6 \cdot (x_{cp})^2$

If x_s = Caterpie: $y' = b_3 + w_3 \cdot x_{cp} + w_7 \cdot (x_{cp})^2$

If x_s = Eevee: $y' = b_4 + w_4 \cdot x_{cp} + w_8 \cdot (x_{cp})^2$

$$y = y' + w_9 \cdot x_{hp} + w_{10} \cdot (x_{hp})^2$$
$$+ w_{11} \cdot x_h + w_{12} \cdot (x_h)^2 + w_{13} \cdot x_w + w_{14} \cdot (x_w)^2$$

y

Training Error = 1.9

Testing Error = 102.3

Overfitting!

train error << test error

Back to step 2: Regularization

model $$y = b + \sum w_i x_i$$

loss function $$L = \sum_n \left(\hat{y}^n - \left(b + \sum w_i x_i \right) \right)^2 \quad \boxed{+\lambda \sum (w_i)^2} \rightarrow \text{Regularization}$$

正则化

The functions with smaller w_i are better

- Smaller w_i means ... smoother

参数越接近 0 → 平滑的function

$$y = b + \sum w_i x_i$$

输入变化 → Output 变化较小

$$y + \sum w_i \Delta x_i = b + \sum w_i (x_i + \Delta x_i)$$

输入变

(对input的变化不敏感)

- We believe smoother function is more likely to be correct

为什么喜欢平滑的function？

Do you have to apply regularization on bias?

Back to step 2: Regularization

$$y = b + \sum w_i x_i$$

$$L = \sum_n \left(\hat{y}^n - \left(b + \sum w_i x_i \right) \right)^2 \quad \boxed{+\lambda \sum (w_i)^2}$$

The functions with smaller w_i are better

- Why smooth functions are preferred?

$$y = b + \sum w_i x_i$$

$+w_i \Delta x_i$ $\quad$ $+\Delta x_i$

- If some noises corrupt input x_i when testing

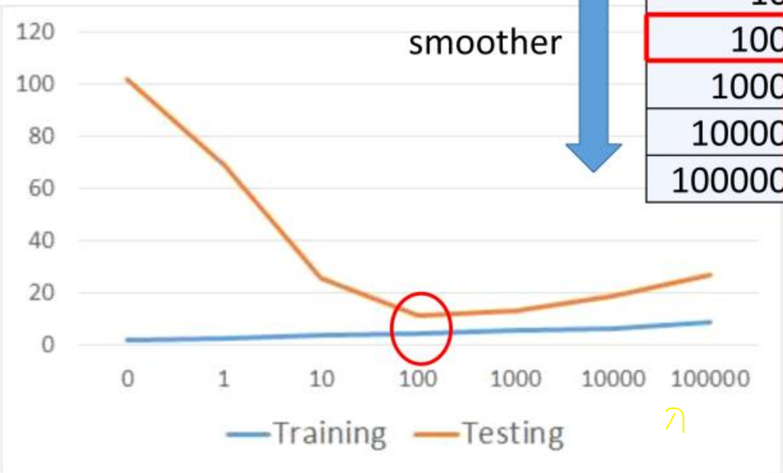
噪音

A smoother function has less influence.

② λ↑，function 越平滑

Regularization

smoother



λ	Training	Testing
0	1.9	102.3
1	2.3	68.7
10	3.5	25.7
100	4.1	11.1
1000	5.6	12.8
10000	6.3	18.7
100000	8.5	26.8

③ function 变平滑

但不要太平滑

如变成水平线

How smooth?

Select λ obtaining the best model

所以要调 λ

① ➢ Training error: larger λ, considering the training error less

↳ 更倾向于考虑 w 本来的值，减少考虑 error

③ ➢ We prefer smooth function, but don't be too smooth.

Conclusion

- Pokémon: Original CP and species almost decide the CP after evolution
  - There are probably other hidden factors
- Gradient descent
  - More theory and tips in the following lectures
- We finally get average error = 11.1 on the testing data
  - How about new data? Larger error? Lower error?
- Next lecture: Where does the error come from?
  - More theory about overfitting and regularization
  - The concept of validation

↳ 验证，校验

Reference

- Bishop: Chapter 1.1

Acknowledgment

- 感謝 鄭凱文 同學發現投影片上的符號錯誤
- 感謝 童寬 同學發現投影片上的符號錯誤
- 感謝 黃振綸 同學發現課程網頁上影片連結錯誤的符號錯誤