Regression

Hung-yi Lee
李宏毅

Regression: Output a scalar

Step 1: Model

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

A set of function

Model

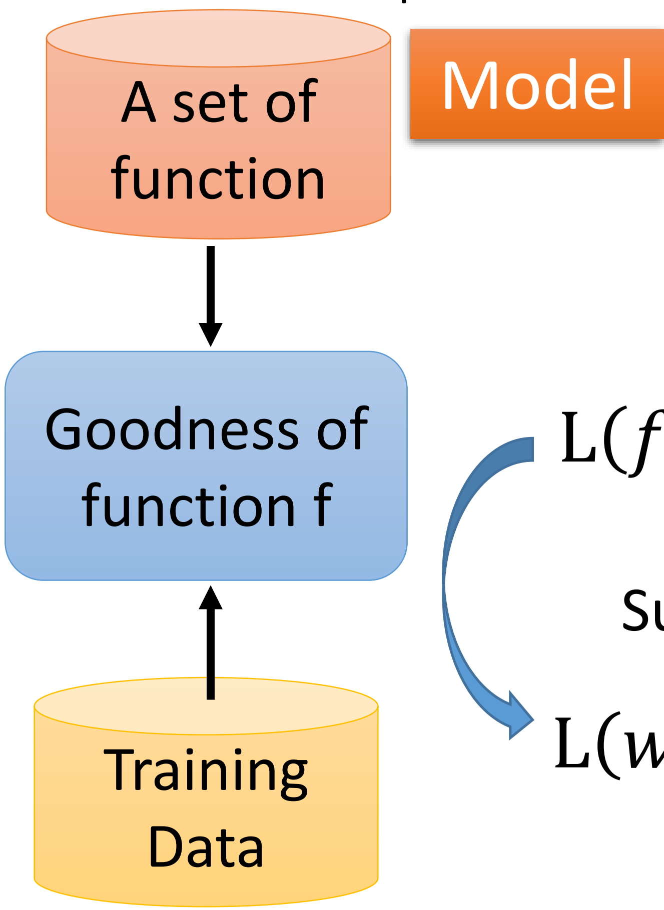
Linear model: $y = b + \sum w_i x_i$

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

w_i: weight, b: bias

Step 2: Goodness of Function

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



Loss function L:

Input: a function,
output: how bad it is

Estimation error

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

Sum over examples

Estimated y based on input function

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

Step 2: Goodness of Function

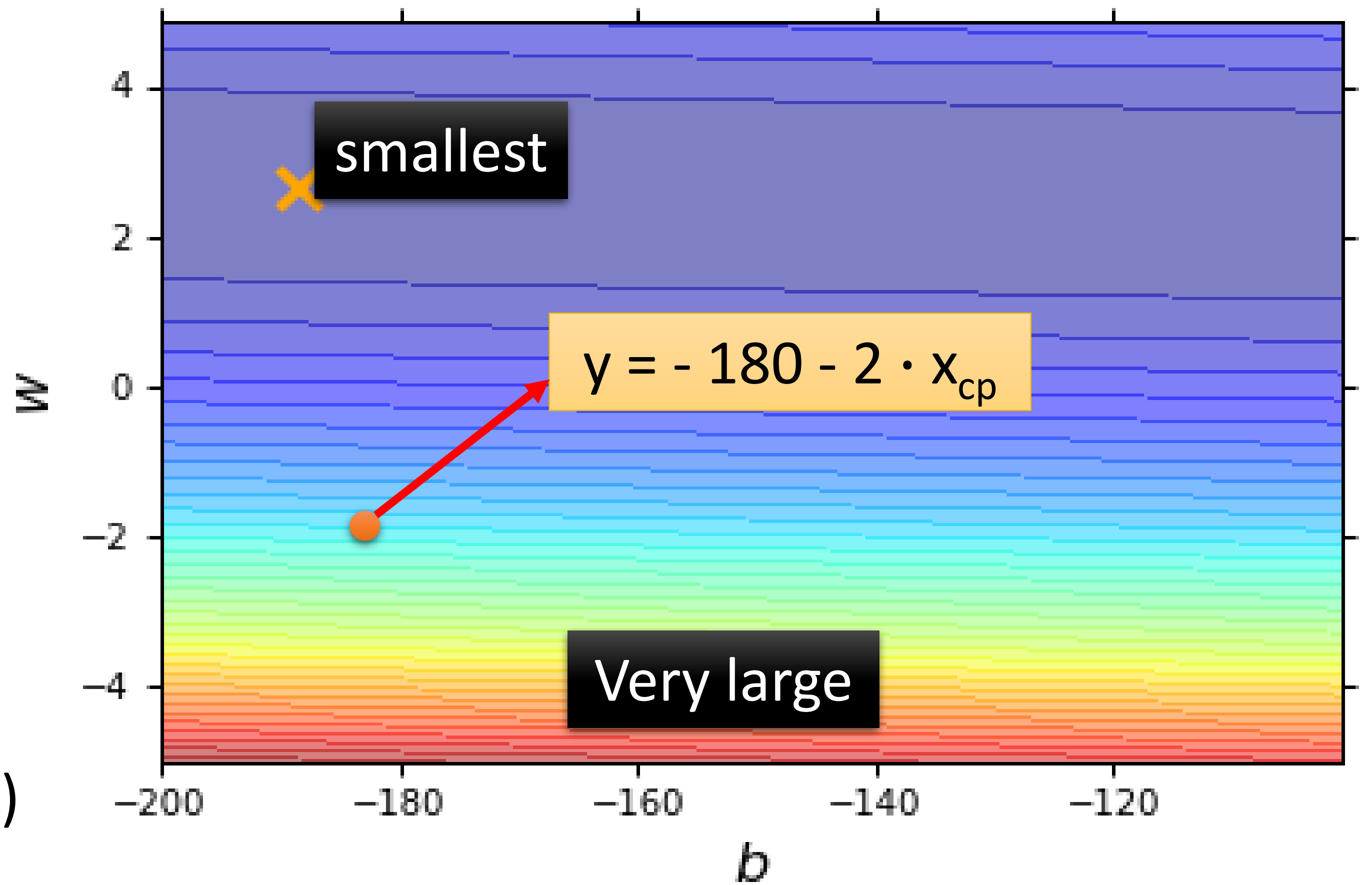
- Loss Function

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

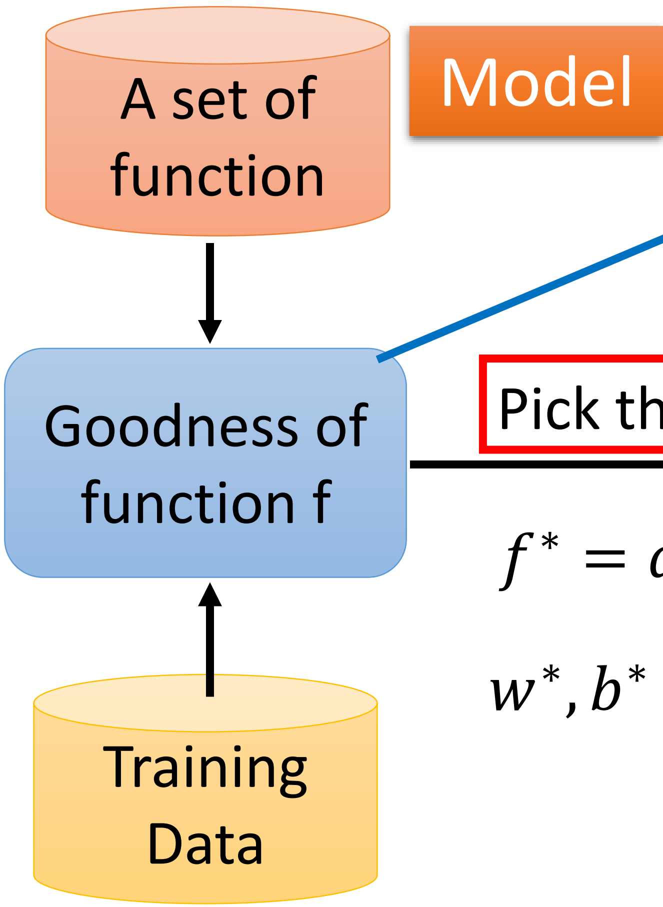
Each point in the figure is a function

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

(true example)



Step 3: Best Function



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

Pick the "Best" Function

$$f^* = arg \min_{f} L(f)$$

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

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

Gradient Descent

Step 3: Gradient Descent

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

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

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

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

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

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

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

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

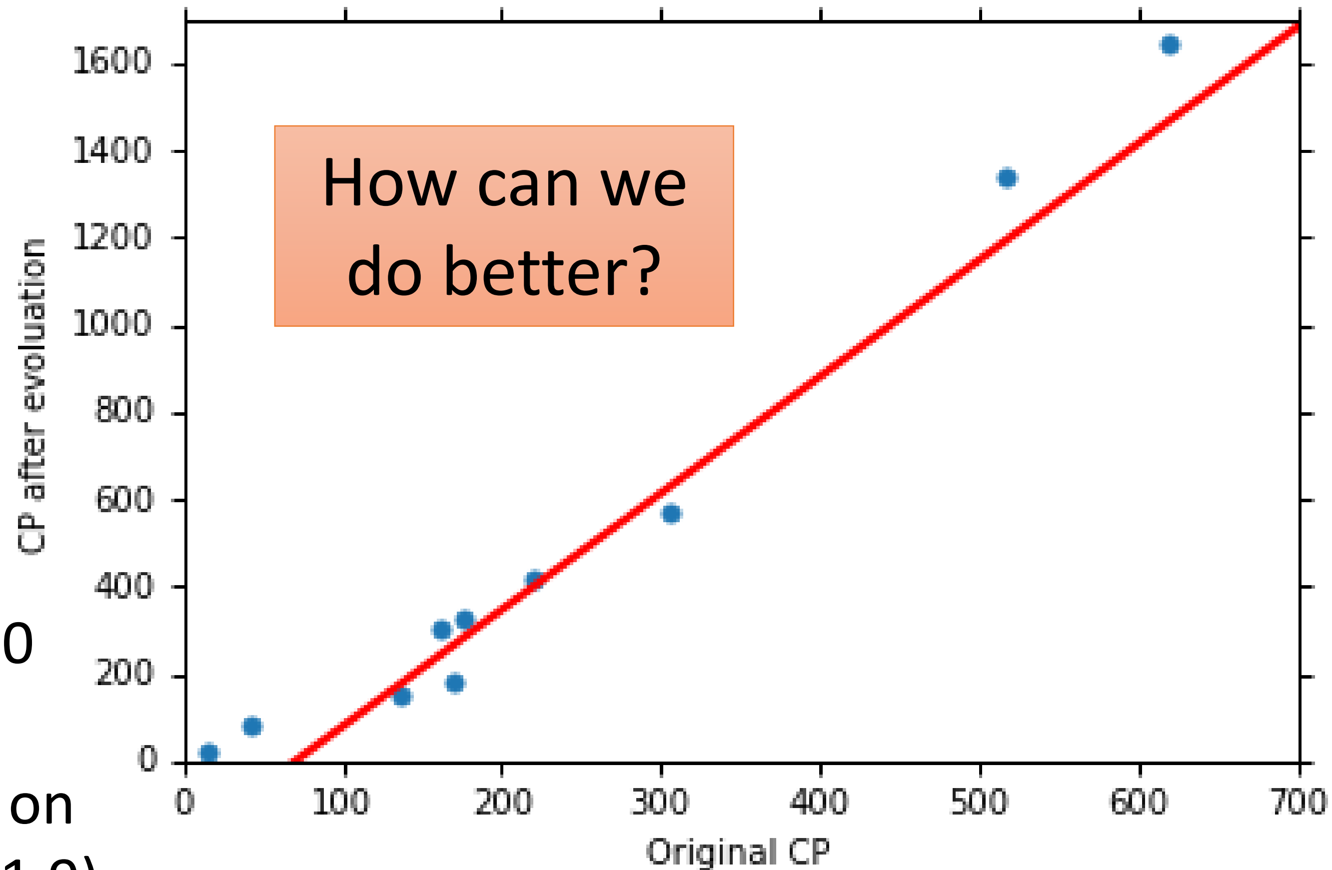
w = 2.7

Average Error on Testing Data

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

> Average Error on Training Data (31.9)

Another 10 pokemons as testing data



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

- Smaller w_i means … smoother

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

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

- We believe smoother function is more likely to be correct

Do you have to apply regularization on bias?

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