Linear Regression – Using Pokemon as Example

Problem to solve: Estimate the Combat Power after a Pokemon evolution

Input: Pokemon (CP before involve, Pokemon height, weight, HP etc)

Output: Pokemon CP after evolution (scalar)

Step1: A set of Functions

- Linear Model Solution Example:

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

We have a set of above functions, infinite number of combinations of w and b (w, b)

Generalize the model (more than one features), it will look like:

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

Step2: Measure Functions

- Define a Loss function L

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

Input: a function

Output: how bad the function is

It's the estimation of error, sum over examples

Step3: Select the Best Function

Select the best function (combinations of w and b) to minimize the Loss function:

$$
\begin{aligned}
f^* &= \underset{f}{\operatorname{argmin}} L(f) \\
w^*, b^* &= \underset{w,b}{\operatorname{argmin}} L(w, b) \\
&= \underset{w,b}{\operatorname{argmin}} \sum_{n=1}^{10} (\hat{y}^n - (b + w * x_{cp}^n))^2
\end{aligned}
$$

Method1: Linear Algebra

Method2: Gradient Descent

Consider Loss function L(w) with parameter w and b, find the best w and b that can minimize L(w)

- L(w, b) must be able to be differentiable
- Don't need to worry about local and global minimum for current linear regression

- Step:

  1) (randomly) pick an initial value

  $$w^0, b^0$$

  2) Compute the following differentiable

  $$
  \frac{dL}{dw}|w = w^0, b = b^0
  $$
  $$
  \frac{dL}{db}|w = w^0, b = b^0
  $$

  If the derivative is negative, higher loss on left hand side and lower loss on right hand side, we wants to find lower loss - increase the value of w

$$w^1 = w^0 - \alpha\frac{dL}{dw}|w = w^0, b = b^0$$
$$b^1 = b^0 - \alpha\frac{dL}{db}|w = w^0, b = b^0$$

learning rate is alpha, higher alpha, higher learning efficiency

how much is the step size? it depends on learning rate (alpha) and initial w

Compute

$$w^2 = w^1 - \alpha\frac{dL}{dw}|w = w^1, b = b^1$$
$$b^2 = b^1 - \alpha\frac{dL}{db}|w = w^1, b = b^1$$

After many iterations, we will find local optimal

Gradient:

$$\triangledown L = [\frac{dL}{dw}, \frac{dL}{db}]^T$$

Example

b = -188.4, w = 2.7 for y = b + w * x

Step 1: Compare different forms of model:

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

Step 2: Measure the performance

	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

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

typora-root-url: ./Image/Screen Shot 2021-07-14 at 1.35.57 PM.jpg

Step 3: Select the best model

Model 3, best performance on testing side, good performance on training

Redesign:

Pokemon spiece can be a factor to CP after evolution? Break down to different spieces.

$$y = b_1 * \alpha(x_s = Pidgey) \\ + w_1 * \alpha(x_s = Pidgey)x_{cp} \\ + b_2 * \alpha(x_s = Weedle) \\ + w_2 * \alpha(x_s = Weedle)x_{cp}$$

Redesign Again:

Other Pokemon related variables? Height? Weight?

$$For x_s = Pidgey : y' = b_1 + w_1 * x_{cp} + w_5 * (x_{cp})^2 \\ y = y' + w_9 * x_{hp} + w_1 0 * (x_{hp})^2 + w_1 1 * x_h + w_1 2 * (x_h)^2 + w_1 3 * x_w + w_1 4 * (x_w)^2$$

Training error: 1.9

Testing Error: 102.3

Back to Step 2: Regularization

Change the Loss function

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

Don't need to consider bias (b) when doing visualization

If we have a relative smooth function, when noise in the training set, it won't have an overly strong impact on the output. (Not so sensitive to noise)

Change lambda:

Higher lambda, more smooth function.

lambda	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

Choose lambda to be 100.

Application in Python