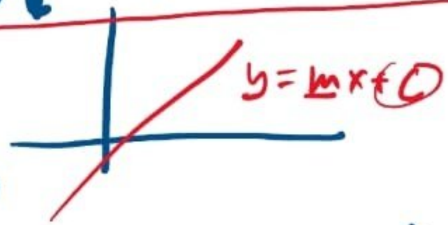
Under/Overfitting

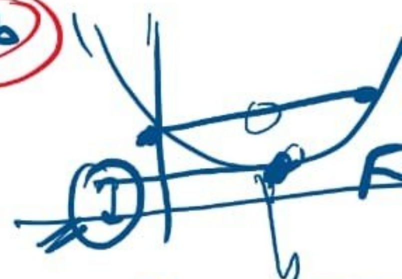
23 December 2025 14:48

lin/log. regression

$f(x) = w_1 x_1 + w_2 x_2 + w_3 x_3 \cdots w_n x_n + b$

y





$f(x) = b + [w_1 \ w_2 \cdots w_n] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$

x_1 → area

1 data (1)

x_n → no of rooms etc

$f(x) = [b \ w_1 \ w_2 \cdots w_n] \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = x^n$

scalar

w^T

row matrix

$f(x) = w^T x^n = \hat{y}^n \approx y^n$

MSE ⇒ $\frac{1}{N} \sum (y^n - w^T x^n)^2$

single value

loss rep f^n

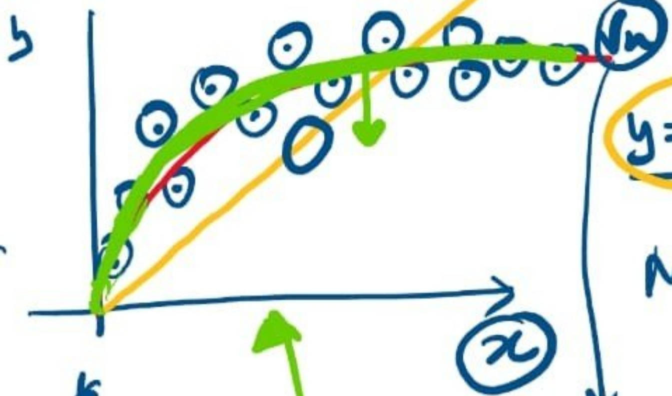
GDesc…

100…

① Underfitting

→ use less for test data itself

① Underfitting



← → useless for test data itself

→ Underfit data / high bias

$y = mx + c$

Non-usable

① More complex model ⇒

② new features] → feature engg.

$(x^k, \sqrt{x^k})$

$\boxed{y^k = w_1 x^k + w_2 \sqrt{x^k}}$ → MSE → $\begin{bmatrix} w_1 \approx 0 \\ w_2 \approx c \end{bmatrix}$ ⇒ Model can't represent structured data

$y^k = [w_1 x^k + w_2 (x^k)^{1/2} + w_3 (x^k)^{1/3} \dots\dots] \Rightarrow$

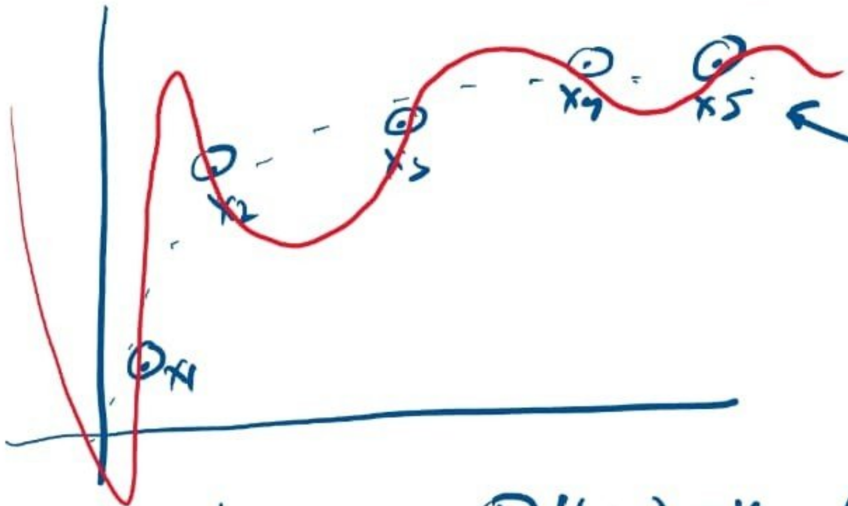
train → (0 : ...) ⇒

② Overfitting ⇒ Model fits on train data ⇒ doesn't fit / give good results ⇒ test data

features
↳ $x^4, x^3, x^2, x^1, \cancel{x^n}$ x^5

$f(x) = w_4x^4 + w_3x^3 + w_2x^2 + w_1x + b$ $+ w_5x^5$

$(w_1, w_2, w_3, w_4, w_5$



① $f(x_1) = y_1$ ② $f(x_2) = y_2$ ③ $f(x_3) = y_3$ ④ $f x_4 = y_4$!!

4 eqn but 5 deg polynomial

∞ soln !! ⇒ problem !!! ⇒

train →

$\frac{1}{N}\sum (y-(w^T x))^2 = 0$

② Overfitting

$[x, x^2, x^3, x^4, x^5, 1]$

$f(x) = \hat{y} = w_1 x_1 + w_2 x^2 + w_3 x^3 + w_4 x^4 + w_5 x^5 + b$

6 quantities in total] MTH113

① $f(x_1) = y_1$ ② $f(x_2) = y_2$ ③ $f(x_3) = y_3$ ④ $f(x_4) = y_4$

→ MLP →

$\boxed{MSE = 0}$ → Straightline of data

[① $MSE_{train} = 0$
② $MES_{test} = $ v. Large !!]

⇒ OVERFIT ⇒

no of pts > 6 ⇒ ∃ unique sol

∃ No sol^n

MSE

$\boxed{MTH113}$

↳ ⊥ subspaces

① Buy/Get → More datapoints

② Regularization :→ $MSE + \lambda \sum |w|^2$

② Regularization ... $MSE + \lambda \sum |w|^2$ ... ⊥ subspaces

② Regularizatn ⇒

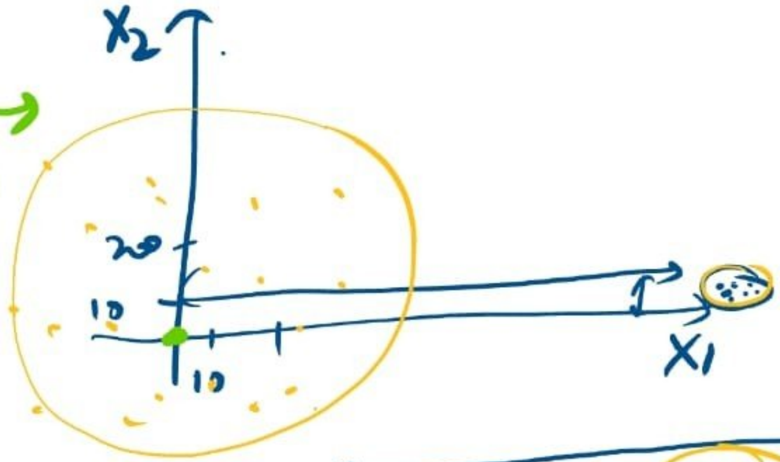
① scaling ⇒ standar-scaling / z-score → normalizatn

setup

$x_1 \rightarrow$ Area of house $\in (1K-10K)$

$x_2 \rightarrow$ No of rooms ⇒ house price

$\in (1, 20)$



mean

$y = w_1 x_1 + w_2 x_2 + b$ ⇒ ① GD → more time ⇒

DW ×10k DW ×10

$$\begin{bmatrix} x_1^1 & x_2^1 \\ x_1^2 & x_2^2 \\ x_1^3 & x_2^3 \\ x_1^4 & x_2^4 \end{bmatrix}$$

$$(x_i^k)_{new} = \frac{(x_i^k)_{old} - \mu_i}{\sigma_i}$$

Std-dev

Mean = 0

Regularizer

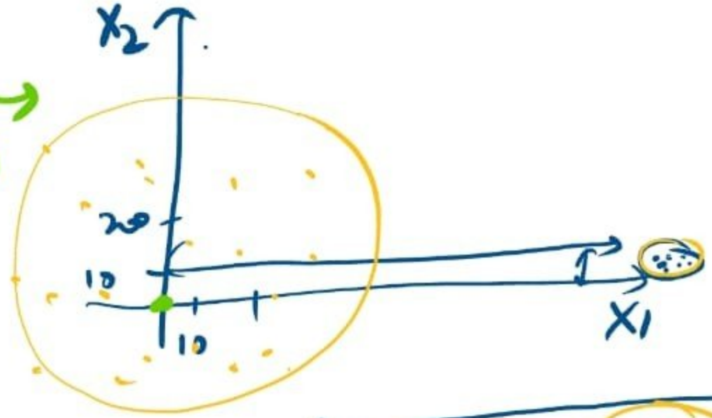
① Scaling ⇒ standar-scaling/z-score → normalized

setup

$x_1 \to$ Area of house $\in$ (1K-10K) ft²

$x_2 \to$ No of rooms ⇒ house price

↳ $\in (1, 20)$



mean

$\hat{y} = w_1 x_1 + w_2 x_2 + b$ → ① GD → more time ⇒

DW ×10K, DW ×10

$$\begin{bmatrix} x_1^1 & x_2^1 \\ x_1^2 & x_2^2 \\ x_1^3 & x_2^3 \\ x_1^4 & x_2^4 \end{bmatrix}$$

$$\boxed{(x_i^k)_{new} = \frac{(x_i^k)_{old} - \mu_i}{\sigma_i}}$$

std-dev

Mean = 0
std = 1

$[w_1\ w_2\ w_3 \cdots w_m] \Rightarrow$ comparable in size

$(w_1, w_2 \cdots w_5) \Rightarrow \to f(x) = x^4 - 1000x^3 + 1$

15:49 Tue, 23 Dec 32%

Recent Pages HOME INSERT DRAW VIEW

Stop

$[x_1^4 \quad x_2^4]$

Mean = 0
std = 1

$[w_1 \; w_2 \; w_3 \cdots w_m] \Rightarrow$ comparable in size

$(w_1 \; w_2 \cdots w_5) \Rightarrow \rightarrow$

$p(x) = x^4 - 1000x^3 + 1$

$p(0) = 1$

$p(1) = -998$

↳ fluctuating rapidly $\Leftarrow$

↳ some $w_i >>>>$ others

↳ $\boxed{loss_{reg} = MSE + \left(\sum |w_i|^2\right) \lambda}$

15:50 Tue, 23 Dec 32%

Recent Pages HOME INSERT DRAW VIEW

Stop

$[x_1^4 \quad x_2^4]$

Mean = 0
std = 1

$[w_1\, w_2\, w_3 \cdots w_m] \Rightarrow$ comparable in size

$(w_1\, w_2 \cdots w_5) \Rightarrow \rightarrow$

$p(x) = x^4 - 1000x^3 + 1$

$p(0) = 1$

$p(1) = -998$

↳ fluctuating rapidly $\Leftarrow$

↳ some $w_i >>>>$ others

↳ $\boxed{loss_{reg} = MSE + \left(\sum |w_i|^2\right) \lambda}$

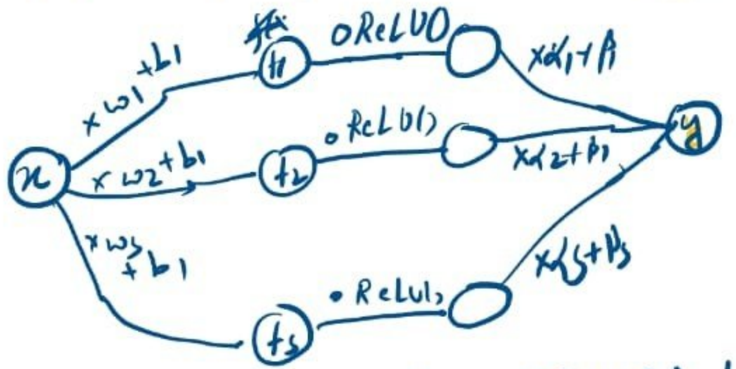
$l^2_{reg} \Rightarrow$

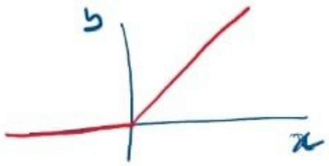
$\Rightarrow l^1 \rightarrow \left(\sum |w_i|\right)$

argmax $\Rightarrow [w_i]_{max}$

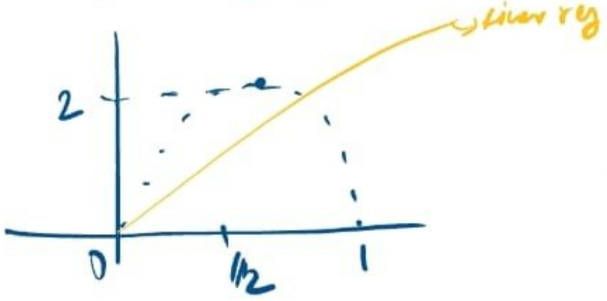
MLP → Multilayer perceptron :- neural-networks

$f(g(x))$

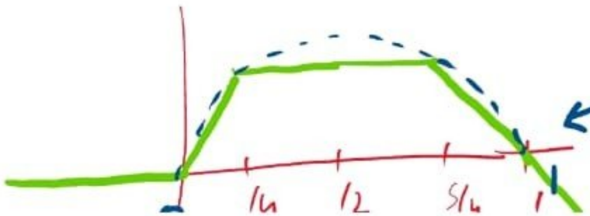




$$y = \alpha_1 \text{Relu}(w_1 x + b_1) + \alpha_2 \text{Relu}(w_2 x + b_2) + \alpha_3 \text{Relu}(w_3 x + b_3)$$



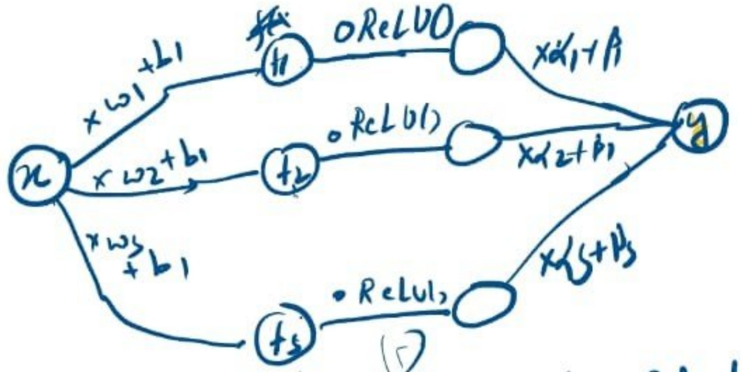
$$\left\{\begin{array}{l} ① \ \text{Relu}(4x) \\ ② \ \text{Relu}(4x-1) \\ ③ \ \text{Relu}(x-3) \end{array}\right] \Rightarrow \left\{\begin{array}{l} \alpha_1 = 1 \\ \alpha_2 = -1 \\ \alpha_3 = -1 \end{array}\right\}$$

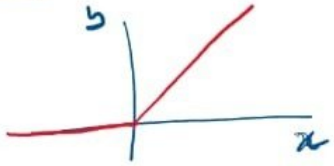


$\rightarrow (\epsilon) \rightarrow (n \in N) \; |f(x) \pm \epsilon|$

MLP → Multilayer perceptron :- neural-networks

$f(g(x))$

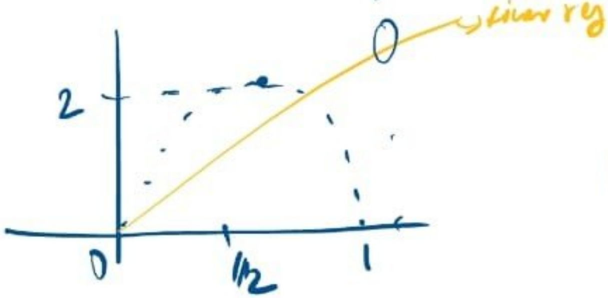




$(\Sigma \text{ Relu } f^n) \times \alpha \Rightarrow$

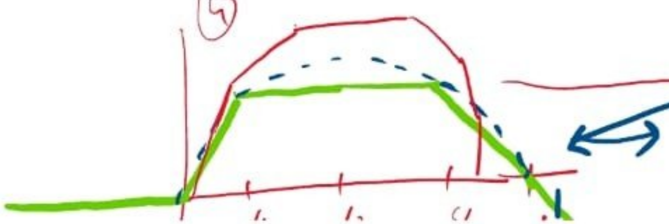
gives piece wise linear graph

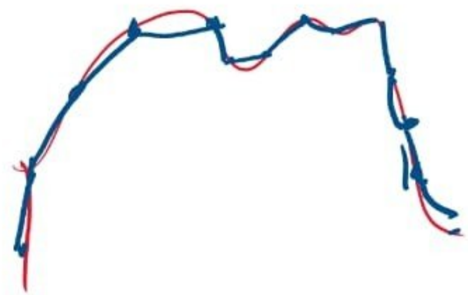
$$y = \alpha_1 \text{Relu}(w_1 x + b_1) + \alpha_2 \text{Relu}(w_2 x + b_2) + \alpha_3 \text{Relu}(w_3 x + b_3)$$



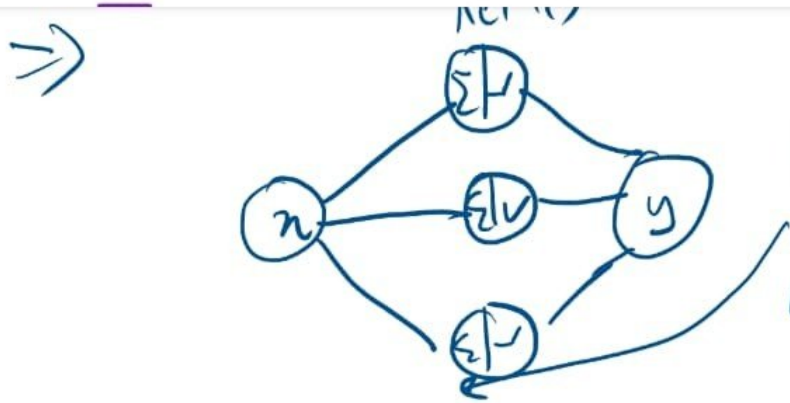
① Relu(2x)
② Relu(4x − 1)
③ Relu(x − 3)

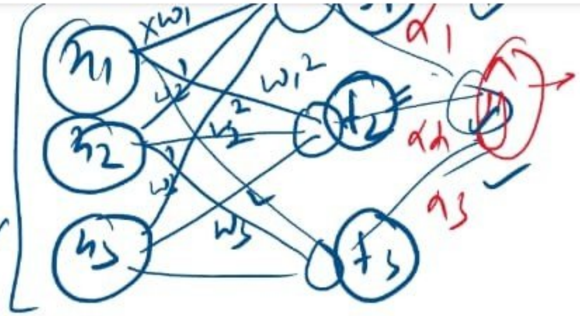
$\Rightarrow \alpha_1 = 1, \; \alpha_2 = -1, \; \alpha_3 = -1$





UNIVERSAL APPROXIMATION





$l = S(f_1\alpha_1 + f_2\alpha_2 + f_3\alpha_3)$

$Relu(w_1^1 x_1 + w_2 x_2 + w_3^1 x_3 + b_1)$

$$\frac{\partial l}{\partial w} = \frac{\partial l}{\partial w_1^1} = \frac{\partial l}{\partial f_1}\,\frac{\partial f_1}{\partial w_1}$$

Back-propagat^n

$$f_1 = Relu(w_1^1 x_1 + w_2^1 x_2 + w_3^1 x_3)$$
$$f_2 = Relu(w_1^2 x_1 + w_2^2 x_2 + w_3^{3} x_3)$$
$$f_3 = Relu(w_1^3 x_1 + w_2^3 x_2 + w_3^3 x_3)$$

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \end{bmatrix} = Relu\left(\begin{bmatrix} w_1^1 & w_1^2 & w_1^3 \\ w_1^2 & w_2^2 & w_3^2 \\ w_1^3 & w_2^3 & w_3^3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}\right)$$

$9 + 3 + 3 = 15$

$\nabla \quad \frac{\partial l}{\partial w}$

α_1

16:18 Tue, 23 Dec 20%

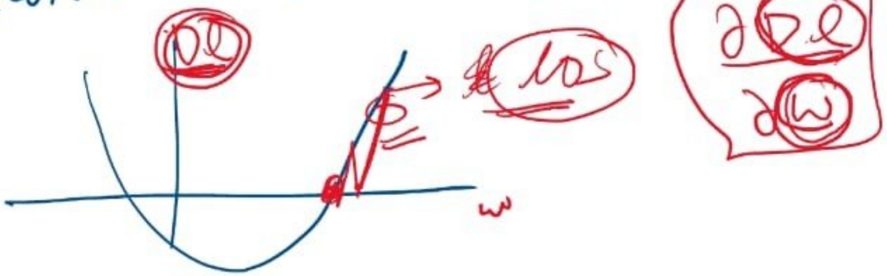
Recent Pages HOME INSERT DRAW VIEW

Stop

$$\boxed{\vec{f} = w\vec{x} + b} \rightarrow 4D$$

$\boxed{\nabla l = 0}$ → Newton-Ralphosis Method

$w_1\ w_2$



deep