19 December 2023 22:02

Deep learning Day-6

* Types of loss function

Regression loss function	Binary Classification loss function	Multiclass Classification loss function
(1) MSE (mean squared error) (2) MAE (mean absolute error) (3) MSLE (mean squared logarithmic error)	(1) Binary cross entropy loss function (2) Hinge loss	(1) Multi class cross entropy loss funⁿ or Categorical cross entropy loss funⁿ (2) Sparse Multi class cross entropy loss function or Sparse categorical cross entropy loss funⁿ
- Relu, leaky relu, P-relu, Elu, Swish	- (1) Sigmoid (2) Tanh..	- Softmax

(1) Regression loss function

(1) MSE (Mean squared error)

$$\boxed{MSE = \frac{1}{n} \sum_{i=1}^{n} (y_a - y_p)^2}$$

→ MSE penalize (increases) big error value so that optimizer function can easily identify big error term.

let's say $y_a = 5$

$$(y_a - y_p)^2$$

$$(5-4)^2 = 1^2 = 1$$

$$(5-3)^2 = 2^2 = 4$$

$$(5-2)^2 = 3^2 = 9$$

— 9-4 = 5

3-2 = 1

→ $y_a = 5$ $y_p = 4.9$ — 0.1

= 4.8 — 0.2

= 4.7 — 0.3

4.6 — 0.4

4.55 — 0.05

4.85 — 0.15

1.8

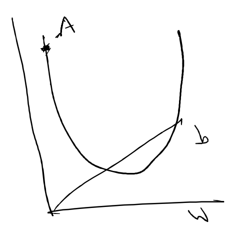
1 Lakh

$(5-4.9)^2 = 0.1^2 = 0.01$

$(5-4.8)^2 = 0.2^2 = 0.04$

$(5-4.7)^2 = 0.3^2 = 0.09$

(5 0.01



$(5-4.9)^2 = 0.3^2 = 0.09$ $= 4.85$ $\boxed{-0.15}$

$(5 - 4.7)$

0.1 — 0.01
0.2 — 0.04
0.3 — 0.09

I   II

→ Draw back of MSE :-

→ Since we are penalizing the big error value, we can not use MSE in case of outlier.

② MAE (mean absolute error)

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_a - y_p|$$

→ We can use MSE in case of outlier.

③ MSLE (mean squared logarithmic error)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_a - y_p)^2$$

$$MSLE = \frac{1}{n}\sum_{i=1}^{n}(\log(y_a) - \log(y_p))^2$$

0

log(0) ×

$$MSLE = \frac{1}{n}\sum_{i=1}^{n}(\log(y_a+1) - \log(y_p+1))^2$$

100   log(100) = =

MSE > MSLE
MAE

② Classification loss function

① Binary Classification

① Binary cross entropy loss function

→ It is also called logloss

It is also called log loss function.

→ Conditions for use

1. Classification → Binary classification
2. activation function in output layer → Sigmoid
3. Dependent variable should be encoded

$$loss = -\frac{1}{n}\sum_{i=1}^{n}[y_a \log y_p + (1-y_a)\log(1-y_p)]$$

(2) Hinge loss

* Conditions for use:-

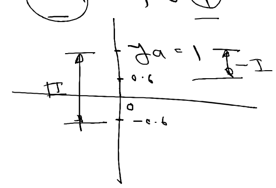
1. Classification → Binary
2. Activation function used in output layer → tanh
3. Dependent variable should be encoded in -1 or 1

$$\boxed{\text{Hinge loss} = \max(0, 1 - y_a y_p)}$$

tanh — Range → -1 to 1

→ (1) $y_a = 1$ $y_p = 0.6$

(2) $y_a = 1$ $y_p = -0.6$



Case I Hinge loss $= \max(0, 1 - y_a y_p)$

$= \max(0, 1 - 1 \times 0.6)$

$$\boxed{\text{Hinge loss} = 0.4}$$

Case II Hinge loss $= \max(0, 1 - 1 \times (-0.6))$

$$\boxed{\text{Hinge loss} = 1.6}$$

tanh — Range - -1 to 1

(1) -0.6 → -1.

tanh — Range — -1 to 1

(1) -0.5 → -1

(2) +0.5 → 1

* multi class classification loss func

(1) multi class cross entropy loss function
(categorical cross entropy loss function)

$$\boxed{\text{loss} = -\sum_{i=1}^{n} y_a \log y_p}$$

→ Conditions for used

(1) Classification → multi class

(2) out put layer activation func → Softmax

(3) Dependent variable should be one hot encoded.

* Draw back

→ In case of multiclass cross entropy loss function dependent variable should be one hot encoded.

A
B
C
D
E

Dependent variable
A
C
D
E
B

One hot encoded Dependent variable — Sparse matrix

A	B	C	D	E
1	0	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1
0	1	0	0	0

(2) Sparse multi class cross entropy loss function

Sparse matrix

A sparse matrix is a special case of a matrix in which the number of zero elements is much higher than the number of non-zero elements. As a rule of thumb, if 2/3 of the total elements in a matrix are zeros, it can be called a sparse matrix.

From <https://www.google.com/search?q=sparse+matrix&rlz=1C1CHBF_enIN1018IN1018&oq=sparse&gs_lcrp=EgZjaHJvbWUqCggBEAAYsQMYgAQyDwgAEEUYORiDARixAxiABDIKCAEQABixAxiABDIKCAIQABixAxiABDIHCAMQABiABDIHCAQQABiABDIKCAUQABixAxiABDIHCAYQABiABDINCAcQABiDARixAxiABDIHCAgQABiABDIHCAkQABiABNIBCTU5NDlqMWoxNagCALACAA&sourceid=chrome&ie=UTF-8>

→ conditions for use

(1) Classification → multi class

(2) out put layer activation → Softmax

(1) Classification → multi class

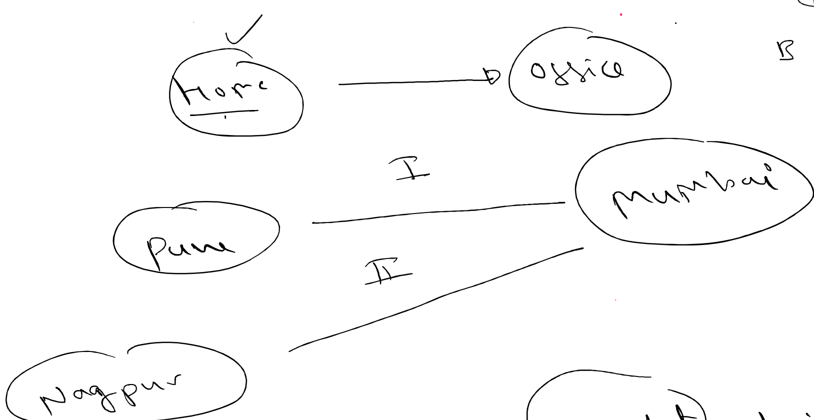
(2) Out put layer activation function → Softmax

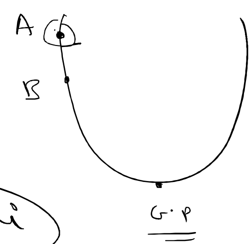
(3) dependent variable should be one hot encoded but loss will be calculated for 1

$$\boxed{\text{loss} = -\sum_{i=1}^{n} y_a \log y_p}$$

1

* **Initializer**





Parameter → Weight, bias, learning rate

0.1

0.001

Initializer

↳ weight..

↳ Distribution

* **Types of Initializer**

(1) He initializer

(2) Glorot initializer

He initializer

- He normal distribution
- He uniform distribution

Glorot initializer

- Glorot normal distribution
- Glorot uniform distribution

Glorot normal distribution

Glorot uniform distribution

→ General thumb rule

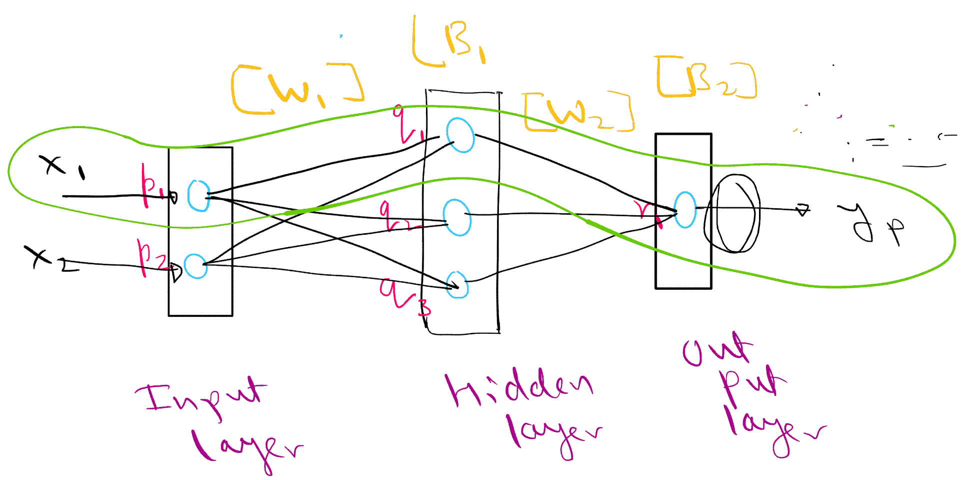
→ If we deal with regression problem he inializer will work good.

→ If we deal with classification problem that time Glorot work well.

	x_1	x_2	y
1	100	200	1
2	300	50	0

0 to 1

0.4



① $x_1 \longrightarrow p_1 \xrightarrow{w'_1} q_1 \xrightarrow{w'_3} r_1 \xrightarrow{b'_2} y_{p_1}$ $10\theta_1$

② $x_1 \longrightarrow p_1 \longrightarrow q_2 \longrightarrow r_1 \longrightarrow y_{p_2}$ $10\theta_2$

$x_1 \longrightarrow p_1 \longrightarrow q_3 \longrightarrow r_1 \longrightarrow y_{p_3}$

$x_2 \longrightarrow p_2 \longrightarrow q_1 \longrightarrow r_1 \longrightarrow y_{p_4}$

$x_2 \longrightarrow p_2 \longrightarrow q_2 \longrightarrow r_1 \longrightarrow y_{p_5}$

$x_2 \longrightarrow p_2 \longrightarrow q_3 \longrightarrow r_1 \longrightarrow y_{p_6}$

$$x_2 \longrightarrow p_2 \longrightarrow q_3 \longrightarrow r_1 \longrightarrow y_{p-6}$$