

## What is Loss Function?

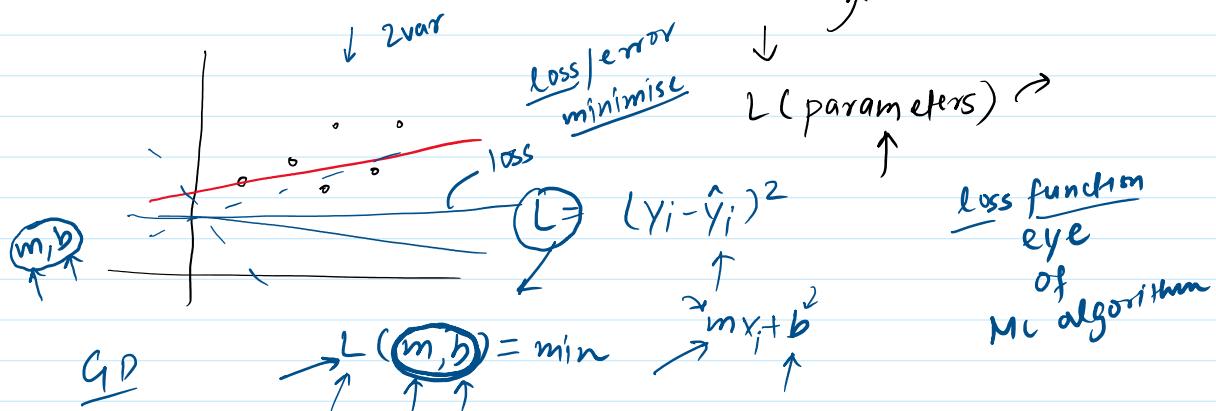
23 March 2022 07:15

Loss function is a method of evaluating how well your algorithm is modelling your dataset.

high → poor

small → great

$$f(x) = x^2 + 2$$



Why is Loss function important?

[You can't improve what you can't measure.]

Peter Drucker

multiplic  
epoche

Loss Function in Deep Learning?

Backprop

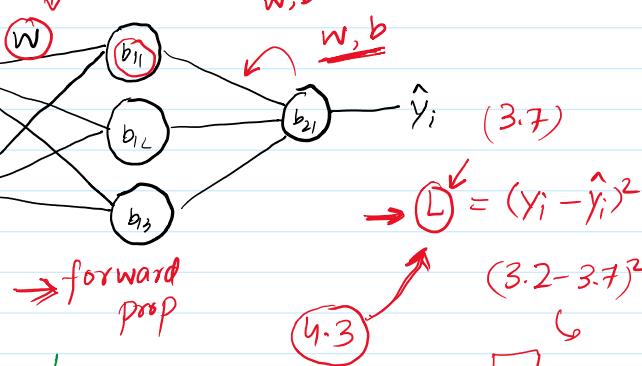
random  
w, b

gradient  
descent

✓ cgpa | iq | package(epa)

	7.1	83
	8.5	91
	6.3	102
	5.1	87
cgpa	3.2	4.5
	6.1	2.7

7.1  
83  
iq



Keras

- object detection
- classification
- Regression
- object detection
- classification
- Regression

{ mse  
mae  
huber loss

Loss functions in DL

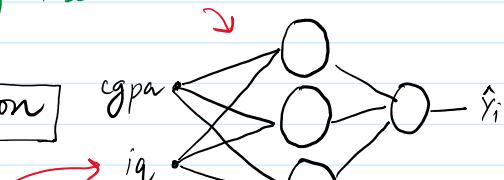
{ classification  
binary crossentropy  
categorical cross entro  
hinge loss

Autoencoders  
+ KL divergence

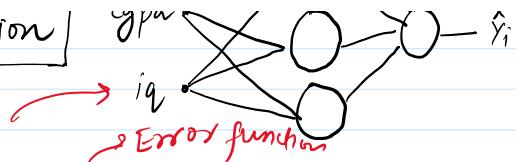
Embedding  
Tripled  
loss

GAN  
+ discriminator  
loss  
minmax  
gan loss

LOSS Function vs Cost Function



## Loss Function vs Cost Function



Loss function → single training eg

$$y_i = 6.3 \quad \hat{y}_i = 6.1$$

$$(y_i - \hat{y}_i)^2$$

$$(6.3 - 6.1)^2 =$$

$$\frac{1}{4} [(6.1 - 6.3)^2 + (4.1 - 4)^2 + (3.5 - 3.7)^2 + (7.2 - 7)^2] = CF$$

batch Cost function

$$\frac{1}{n} \sum (y_i - \hat{y}_i)^2$$

$$\textcircled{5} \text{ lpa } \sqrt{\sum (y_i - \hat{y}_i)^2}$$

$$(lpa)^2$$

### 1. Mean Squared Error (MSE)

squared loss L2 loss

$$(y_i - \hat{y}_i)^2$$

(true - predict)<sup>2</sup>

$$(6.3 - 6.1)^2 =$$

$$\boxed{(y_i - \hat{y}_i)^2}$$

quadratic punish

magnify

$$(y_i - \hat{y}_i)^2$$

$$(W, b) \rightarrow \textcircled{1} \min$$

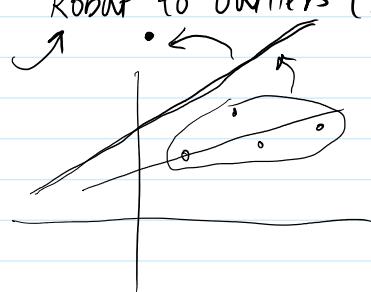
#### Advantages

- 1) Easy to interpret
- 2) Differentiable (GD)
- 3) 1 local minima

$$CF = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2$$

#### Disadvantage

- 1) Error unit (squared) → diff
- 2) Robust to Outliers (Not)



$$\boxed{mse}$$

### 2. Mean Absolute Error (MAE) → L1 loss

$$L = |y_i - \hat{y}_i|$$

$$2 \mid \text{abs}$$

$$C = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|^2$$

punish

$$\mid \text{true - predict} \mid =$$

$$csp | iq | pack \ lpa \mid (y_i - \hat{y}_i) \mid$$

$$\hookrightarrow lpa$$

$$C = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

true-predict  
punish

Advantages

- 1) Intuitive and easy
- 2) Unit  $\rightarrow$  same  $-y$
- 3) Robust to outliers

Disadvantage

Not differentiable

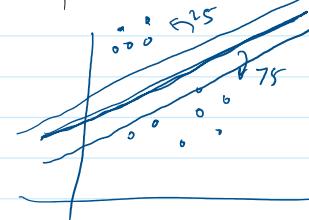
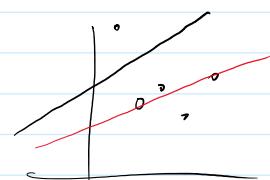
$\nabla L$   $\approx$  differentiable  
Subgradient

mse - outliers ✓  
mae - normal point ✓

3. Huber Loss ✓

$$L = \begin{cases} \frac{1}{2} (y - \hat{y})^2 & \text{for } |y - \hat{y}| \leq \delta \\ \delta |y - \hat{y}| - \frac{1}{2} \delta^2 & \text{otherwise} \end{cases}$$

mae



4. Binary Cross Entropy

→ classification ✓  
→ Two classes ✓

100 days  
cgpa | iq | placement

8	80
7	70
6	60
0	0

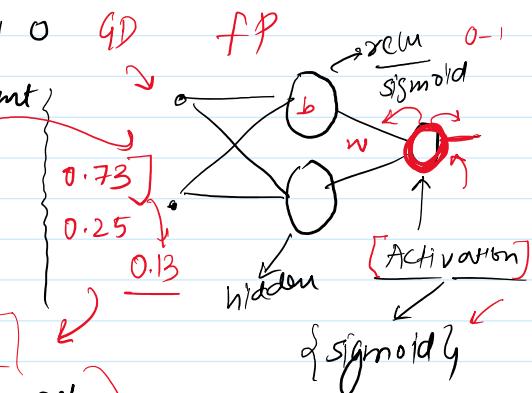
1
0
0

$$\text{LOSS function} = -y \log(\hat{y}) - (1-y) \log(1-\hat{y})$$

Keras

$$0.12 = - (1-0) \log(1-0.25) - 1 \log(0.75)$$

$$\text{cost function} = -\frac{1}{n} \left[ \sum_{i=1}^n y_i \log \hat{y}_i + (1-y_i) \log(1-\hat{y}_i) \right]$$



maximum likelihood

Logistic Reg

Advantage

\* Differentiable

D/advan

→ multi local minima  
→ Intuitive

5. Categorical Cross Entropy [used in Softmax Regression]

→ multi-class classifications ✓

OHE

cgpa	iq	placed?	Yes	No	Maybe
------	----	---------	-----	----	-------

5. Categorical Cross Entropy [Used in softmax regression]

→ Multi-class classification

1 point

$$L = - \sum_{j=1}^K y_j \log(\hat{y}_j)$$

Where  $K$  is # classes in the data

cgpa	iq	placed?	Yes	No	Maybe
8	80	Yes 1	1	0	0
6	60	No 2	0	1	0
7	70	Maybe 3	0	0	1

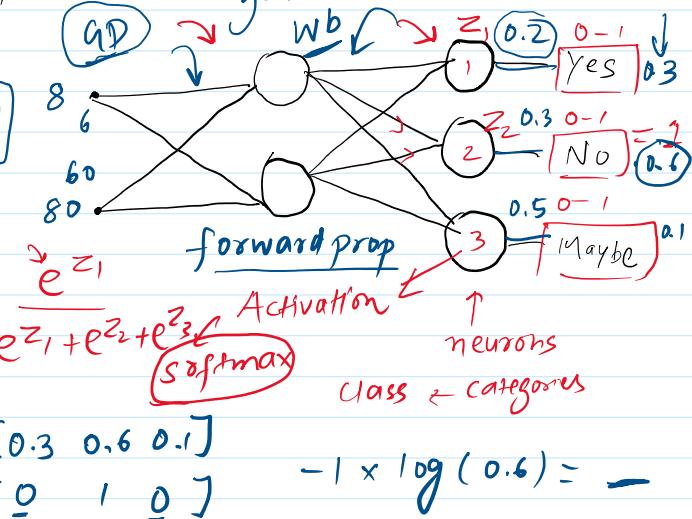
1 point

$$L = - (y_1 \log(\hat{y}_1) + y_2 \log(\hat{y}_2) + y_3 \log(\hat{y}_3))$$

$$[0.2 \ 0.3 \ 0.5] \xrightarrow{e^{z_1} + e^{z_2} + e^{z_3}} f(z) = \frac{e^{z_1}}{e^{z_1} + e^{z_2} + e^{z_3}}$$

$$[1 \ 0 \ 0] \xrightarrow{e^{z_2}} \text{Activation}$$

$$L = -1 \times \log(0.2) \xrightarrow{e^{z_1} + e^{z_2} + e^{z_3}} -1 \times \log(0.6) = -$$



### Sparse Categorical Cross Entropy

(L) ~

$$-1 \times \log(0.1)$$

SCE

Loss function  
fast

cgpa | iq | placed OHE

① 1/2/3	7	70	Yes	1
②	8	80	No	2
③	6	60	Maybe	3

$$[0.1 \ 0.4 \ 0.6]$$

$$[0.6 \ 0.2 \ 0.2]$$

OHE — fast

SCE

$$L = - \sum_{j=1}^K y_j \log(\hat{y}_j)$$

Cost function

$$C = -\frac{1}{n} \sum_{i=1}^n \sum_{j=1}^K y_{ij} \log(\hat{y}_{ij})$$

