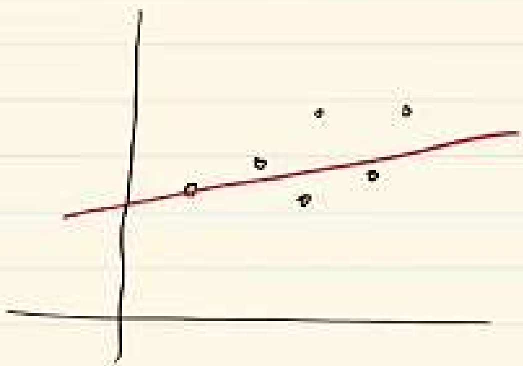
Loss function is a method of evaluating how well your algorithm is modelling your dataset.

- *High means Poor Model*
- *Low means great Model*



$L(\text{parameters})$

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

$mx_i + b^2$

$L(m, b) = \min$

GD

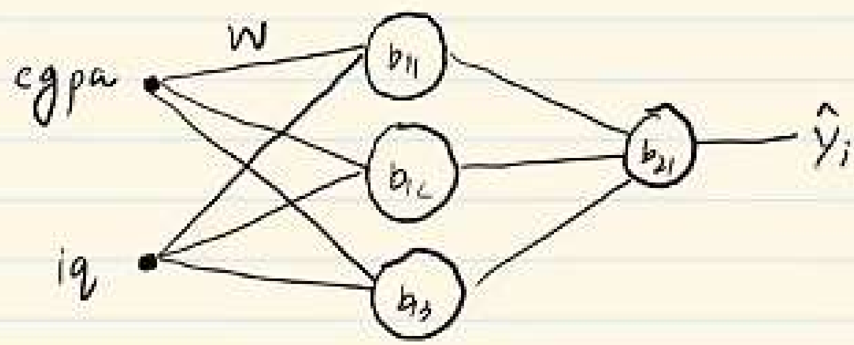
Why is Loss function important?

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

Peter Drucker

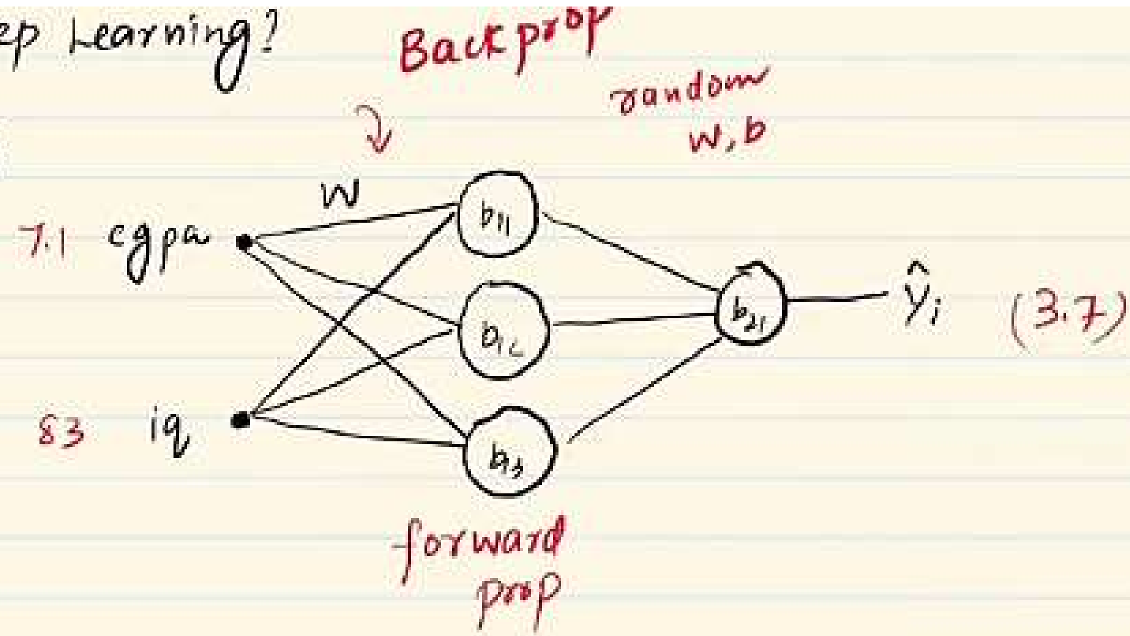
Loss Function in Deep Learning? Backprop

cgpa	iq	package(lpa)
7.1	83	3.2
8.5	91	4.5
6.3	102	6.1
5.1	87	2.7
...	...	...
...	...	...



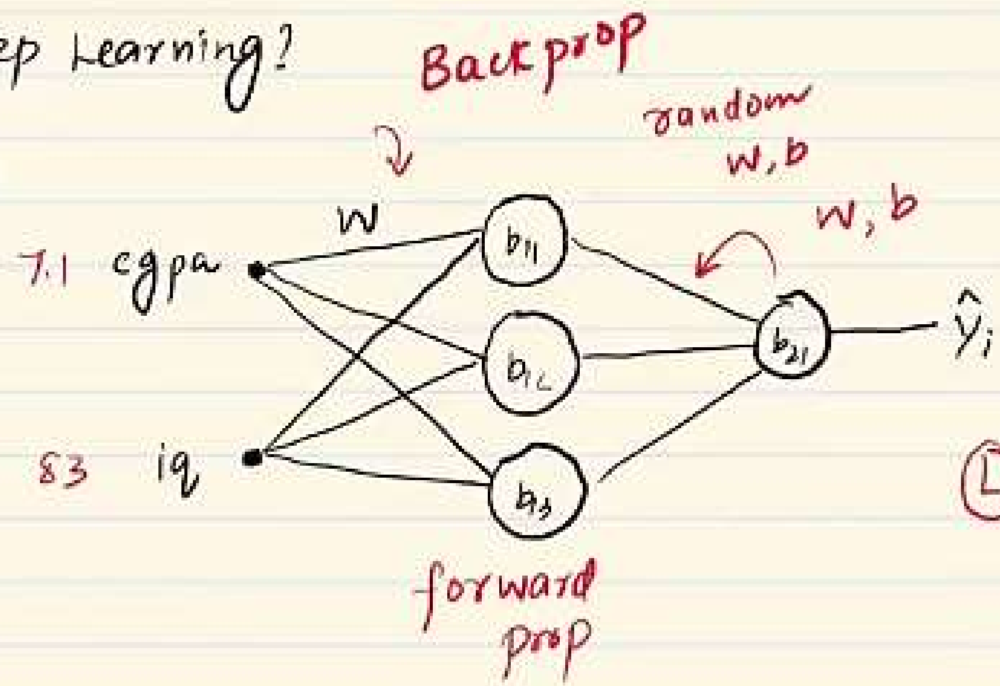
Loss Function in Deep Learning?

cgpa	iq	package(lpa)
7.1	83	3.2
8.5	91	4.5
6.3	102	6.1
5.1	87	2.7
..	..	..
..	..	..
..	..	..



Loss Function in Deep Learning?

cgpa	iq	package (lpa)
7.1	83	3.2
8.5	91	4.5
6.3	102	6.1
5.1	87	2.7
..	..	..
..	..	..
..	..	..



(3.7)

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

$$(3.2 - 3.7)^2$$

epochs

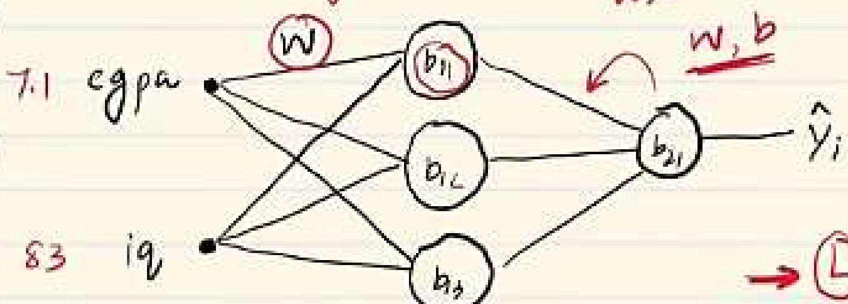
Loss Function in Deep Learning?

Backprop

random w,b → gradient descent

✓ cgpa | iq | package(lpa)

cgpa	iq	package(lpa)
7.1	83	3.2
8.5	91	4.5
6.3	102	6.1
5.1	87	2.7
..	..	..
..	..	..
..	..	..



(3.7)

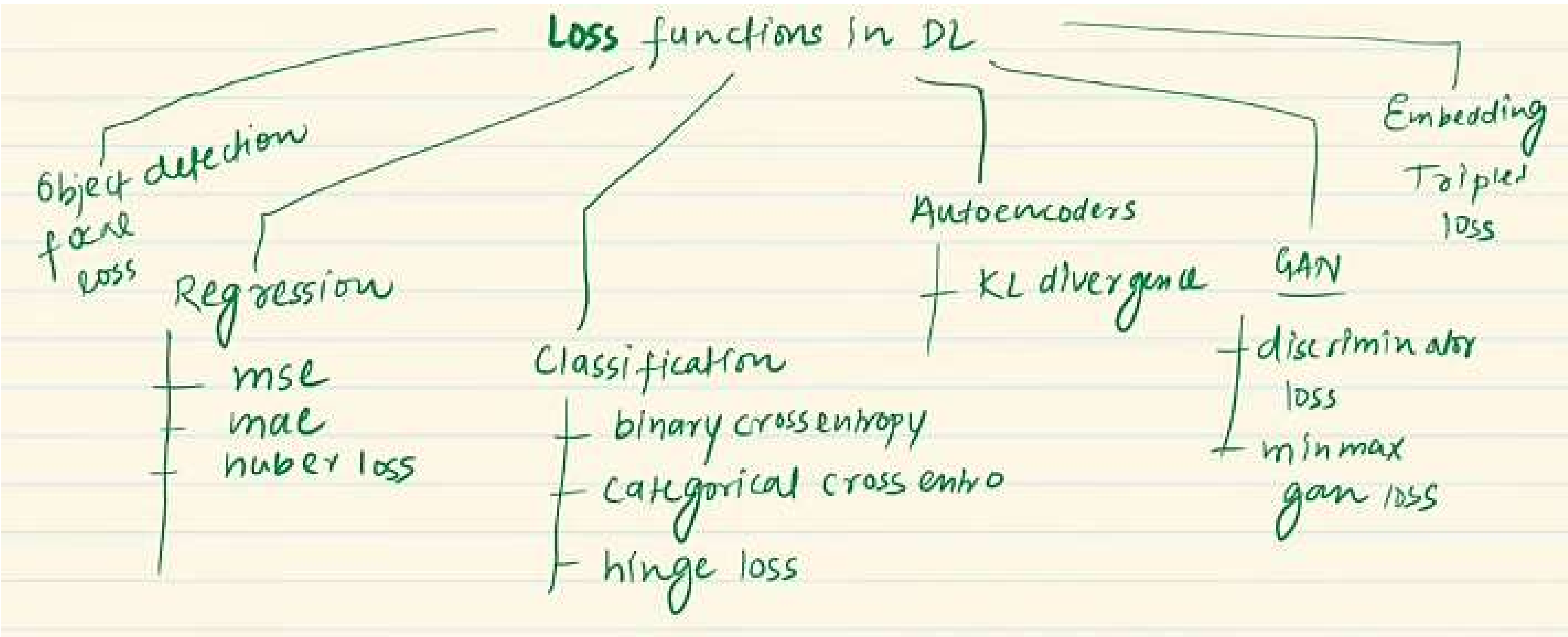
$$L = (y_i - \hat{y}_i)^2$$

$$(3.2 - 3.7)^2$$

→ forward prop

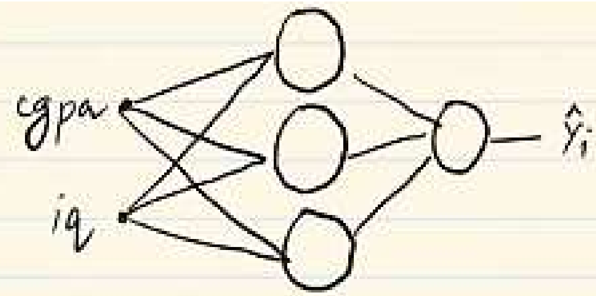
4.3

6

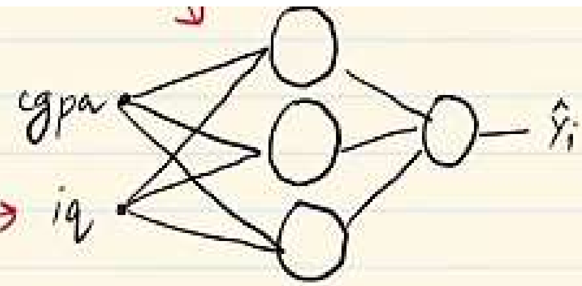

Loss functions in DL
Object detection
focal loss
Regression
mse
mae
huber loss
Classification
binary crossentropy
categorical cross entro
hinge loss
Autoencoders
KL divergence
GAN
discriminator loss
minmax gan loss
Embedding
Triplet loss

$\boxed{\text{Loss Function Vs Cost Function}}$

cgpa	iq	package (y_i)	Prediction ($\hat{y}_i$)
6.3	100	6.3	6.1
7.1	91	4.1	4
8.5	83	3.5	3.7
9.2	102	7.2	7



$\boxed{\text{Loss Function vs Cost Function}}$



cgpa	iq	package (y_i)	Prediction ($\hat{y}_i$)
6.3	100	6.3	6.1
7.1	91	4.1	4
8.5	83	3.5	3.7
9.2	102	7.2	7

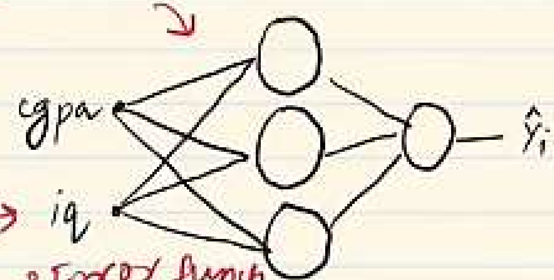
Loss function → single training eg

$y_1 = 6.3 \quad \hat{y}_1 = 6.1$

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

$(6.3 - 6.1)^2 =$

Loss Function vs Cost Function



Error funcn

cgpa	iq	package (y_i)	Prediction ($\hat{y}_i$)
6.3	100	6.3	6.1
7.1	91	4.1	4
8.5	83	3.5	3.7
9.2	102	7.2	7

Loss function → single training eg

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

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

$(6.3 - 6.1)^2 =$

batch Cost function

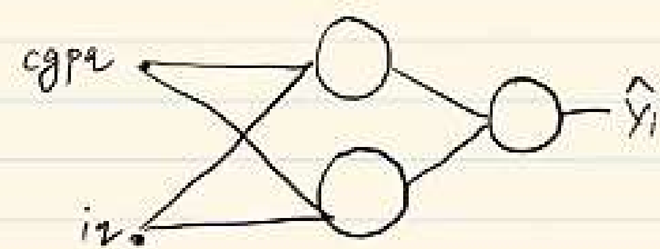
$\frac{1}{n} \sum (y_i - \hat{y}_i)^2$

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

1. Mean squared Error (MSE)

squared loss L2 Loss

cgpa	iq	package (y_i)	Prediction ($\hat{y}_i$)
6.3	100	6.3	6.1
7.1	91	4.1	4
8.5	83	3.5	3.7
9.2	102	7.2	7



1. Mean Squared Error (MSE)

squared loss L2 Loss

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

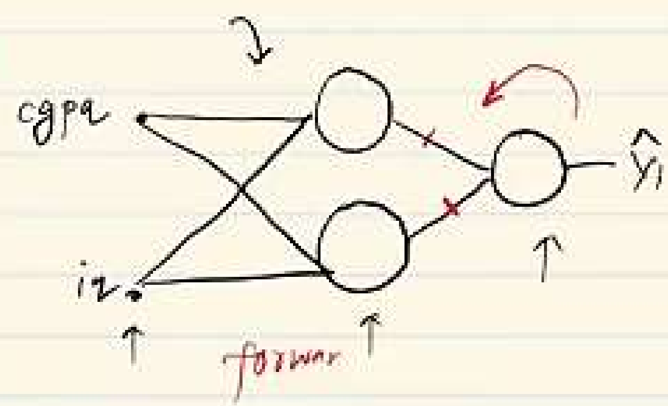
Advan DisAdv

(true - predict)^2

$(6.3 - 6.1)^2 = -$

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

cgpa	iq	package (y_i)	Prediction ($\hat{y}_i$)
6.3	100	6.3	$\boxed{6.1}$
7.1	91	4.1	4
8.5	83	3.5	3.7
9.2	102	7.2	7



1. Mean squared Error (MSE)

squared loss   L2 Loss

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

Advan   DisAdv

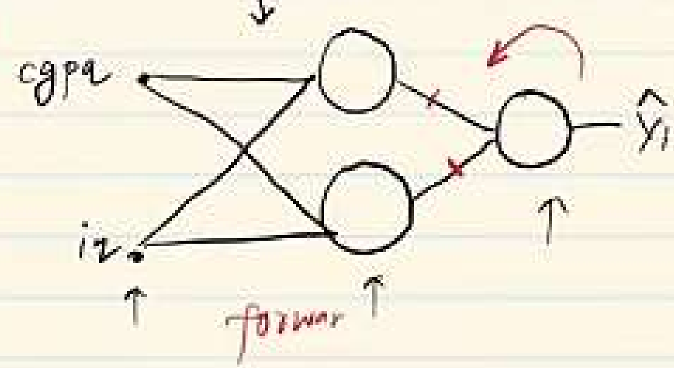
$(\text{true} - \text{predict})^2$

$(6.3 - 6.1)^2 = -$ $\underline{(y_i - \hat{y}_i)^2}$

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

cgpa	iq	package (y_i)	Prediction ($\hat{y}_i$)	$y_i - \hat{y}_i$
6.3	100	6.3	6.1	0.2
7.1	91	4.1	4	0.1
8.5	83	3.5	3.7	-0.2
9.2	102	7.2	7	0.2

overall error



1. Mean Squared Error (MSE)

squared loss L2 Loss

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

$(\text{true} - \text{predict})^2$

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

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

Advan   DisAdv

mae

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

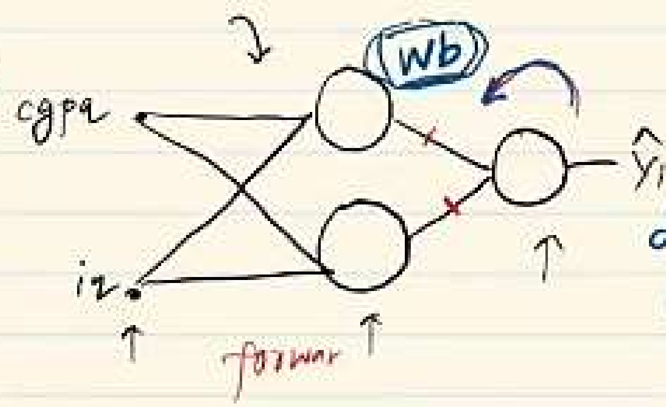
quadratic $\wedge^2$ punish

true − predic → 1 unit → 1 unit
2 unit → 4 uint
4 unit → 16 unit

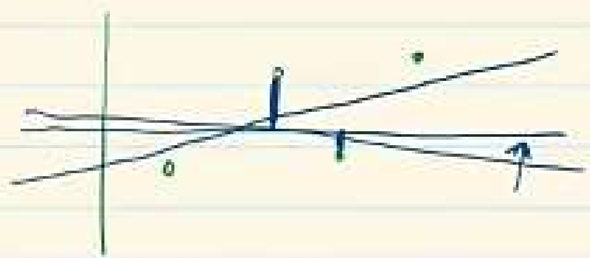
magnify

cgpa	iq	package	Prediction	$y_i - \hat{y}_i$
6.3	100	6.3	6.1	0.2
7.1	91	4.1	4	0.1
8.5	83	3.5	3.7	−0.2
9.2	102	7.2	7	0.2

overall error

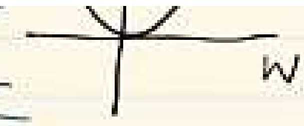


drastic
big



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

$(w, b) \rightarrow L$ min

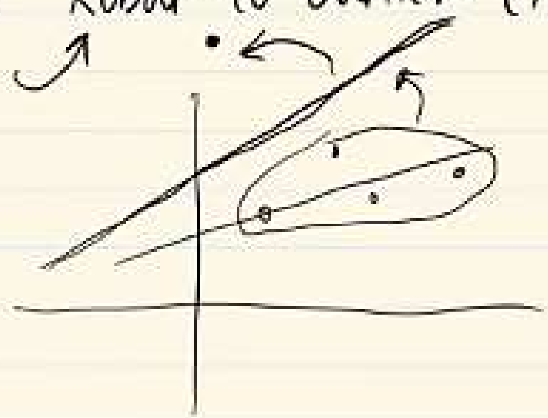


Advantages

1) Easy to intrepret ↗
2) Diffrentiable (GD)
3) 1 local minima

Disadvantage

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



$\boxed{\text{mse}}$

1. Mean squared Error (MSE)

squared loss   L2 Loss

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

(true - predict)^2

$(6.3 - 6.1)^2 = -$

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

Advan   DisAdv

mae

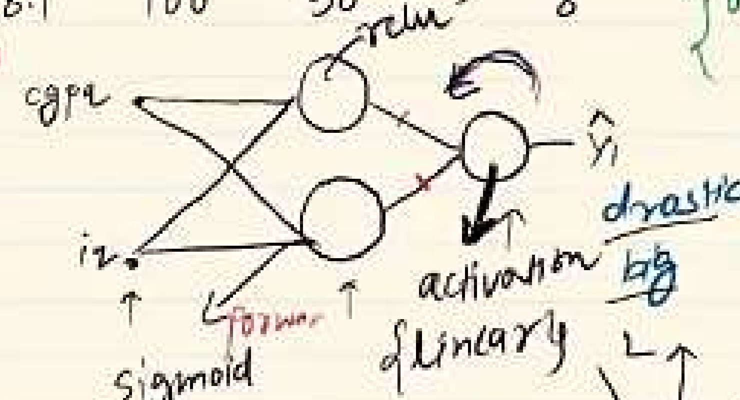
$\boxed{(y_i - \hat{y}_i)^2}$

quadratic ^{2}

punish

cgpa	iq	package	Prediction	$(y_i - \hat{y}_i)$
6.3	100	6.3	6.1	0.2
7.1	91	4.1	4	0.1
8.5	83	3.5	3.7	-0.2
9.2	102	7.2	7	0.2
8.1	100	50	8	

overall error



$(42)^2$

drastic

big

{linear}

true - predict → 1 unit → 1 unit

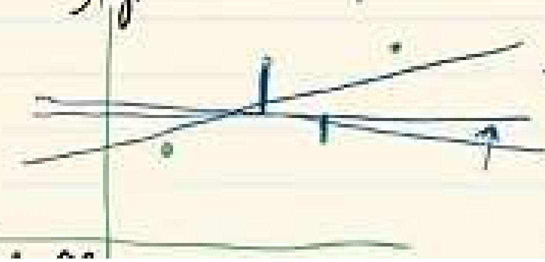
2 unit → 4 unit

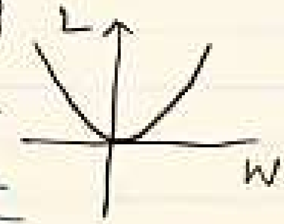
4 unit → 16 unit

magnify

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

$(w, b) \rightarrow L$ min





2 lpa

2. Mean Absolute Error (MAE) → L1 loss

CSpl iql pace lpz $|(y_i - \hat{y}_i)|$

↳ lpa

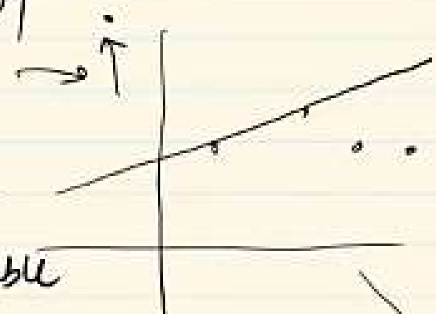
$$L = |y_i - \hat{y}_i|$$

2|-abs

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

|true - predict|

punish



Advantages

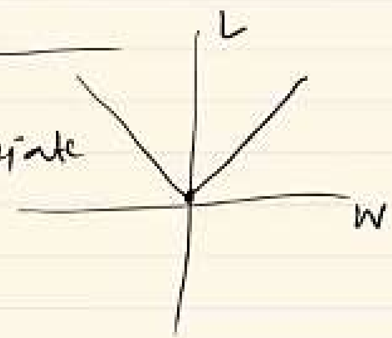
1) Intuitive and easy
2) Unit → same - y
3) Robust to outliers

Disadvange

1) Not differentiable

↳ GD → differentiate

Subgradient

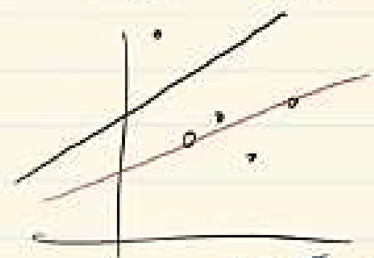


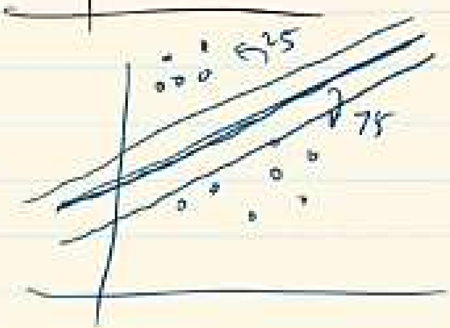
3. Huber Loss ✓

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

mse — outliers ✓

mae — normal points ✓





Subgrad

mae huber mse

3. Huber Loss ✓

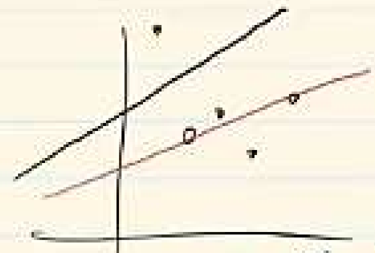
mse   hyper

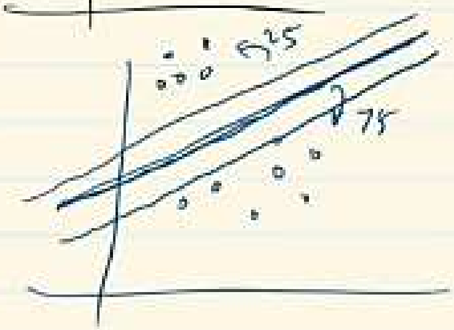
$$L = \begin{cases} \frac{1}{2}(y-\hat{y})^2 & \underbrace{\text{for } |y-\hat{y}| \le \delta}_{} \\ \delta|y-\hat{y}| - \frac{1}{2}\delta^2 & \underbrace{\text{otherwise}}_{} \end{cases}$$

mae

mse — outliers ✓

mae — normal points ✓





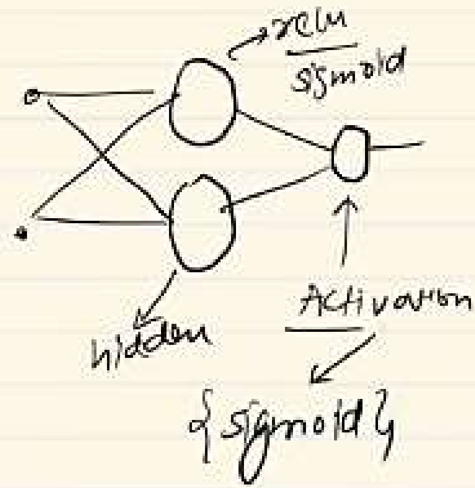
4. Binary Cross Entropy

→ classification

→ Two classes

1 0

cgpa	iq	placement
8	80	1
7	70	0
6	60	0



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

$y \to$ actual value / target

$\hat{y}$ – NN prediction

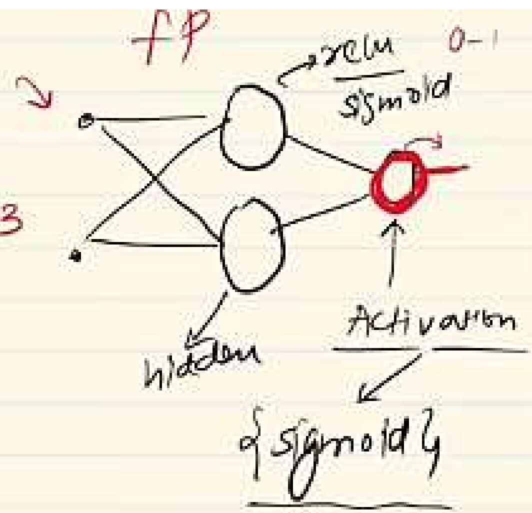
4. Binary Cross Entropy

→ classification
→ Two classes

1 0

cgpa	iq	placement
8	80	1
7	70	0
6	60	0

0.73



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

$y \to$ actual value / target

$\hat{y}$ – NN prediction

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

100 days

4. Binary Cross Entropy

→ classification ✓
→ Two classes ✓

cgpa | iq | placement

8	80	1
7	70	0
6	60	0

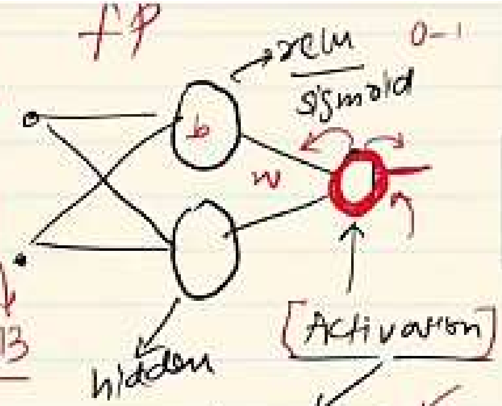
1 0

0.12

0.73
0.25
0.13

GD

FP



{sigmoid}

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

Keras

$y \to$ actual value / target

$\hat{y}$ - NN prediction

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

$-1\log(0.75)$ -0.12

$-1\log(0.73)$

$-1 \times -0.13 = 0.13$

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

maximum likeli

[Logistic Reg]

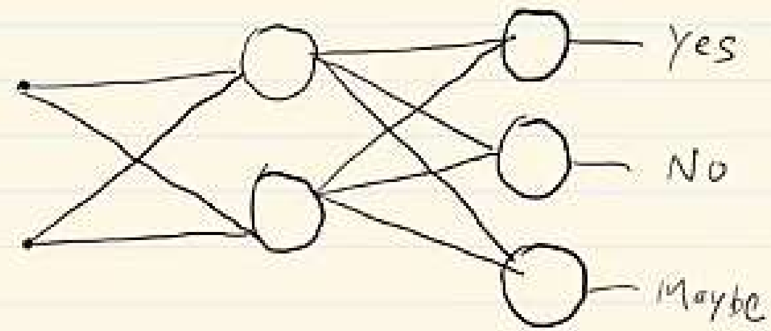
5. Categorical Cross Entropy [used in Softmax Regression]

→ [Multi-class] {classification}

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

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



5. Categorical Cross Entropy [used in Softmax Regression]

→ [Multi-class] {classification}

1 point

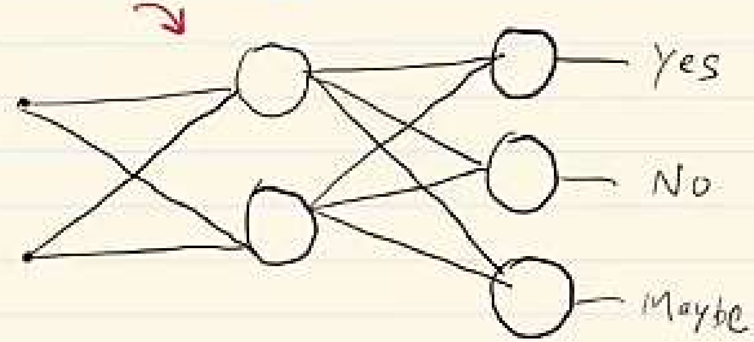
$$\boxed{L = -\sum_{j=1}^{k} y_j \log(\hat{y}_j)}$$

where k is # classes in the data
↳ 3

1 point

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

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



5. Categorical Cross Entropy [used in Softmax Regression]

→ [Multi-class] {classification}

1 point

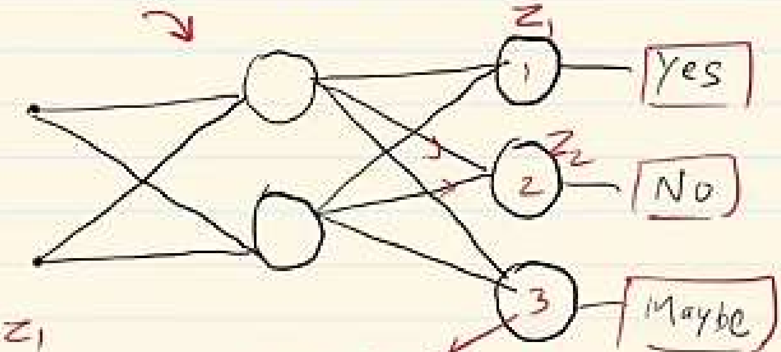
$$\boxed{L = -\sum_{j=1}^{k} y_j \log(\hat{y}_j)}$$

where K is # classes in the data ↳ 3

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

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



$$\frac{e^{z_3}}{e^{z_1}+e^{z_2}+e^{z_1}}$$

$$f(z) = \frac{e^{z_1}}{e^{z_1}+e^{z_2}+e^{z_3}}$$

softmax

Activation

neurons

$$\frac{e^{z_2}}{e^{z_1}+e^{z_2}+e^{z_3}}$$

class ← categories

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
↳ 3

1 point

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

[0.2 0.3 0.5]

[1 0 0]

$$\frac{e^{z_3}}{e^{z_1}+e^{z_2}+e^{z_1}}$$

$$\frac{e^{z_2}}{e^{z_1}+e^{z_2}+e^{z_3}}$$

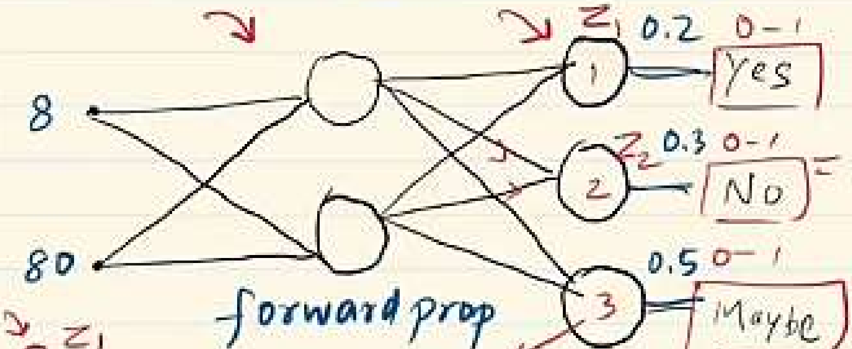
$$f(z) = \frac{e^{z_1}}{e^{z_1}+e^{z_2}+e^{z_3}}$$

softmax

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

OHE

Yes	No	Maybe	
1	0	0	Yes
0	1	0	No
0	0	1	Maybe



forward prop

Activation

neurons

class ← categories

Reg → mse

↳ outliers – mae ↳ huber loss

Classification → binary → bce

↳ multi → CCE

↳ SCE