機器與深度學習基礎知識初探
-Loss Function

黃志勝 Chih-Sheng (Tommy) Huang

義隆電子人工智慧研發部

國立陽明交通大學AI學院合聘助理教授

Introduction

- In learning algorithm, there is an assumption, which may accompany with an object function.

  K-mean: minimizing the mean of square error between data and centers.

  PCA: **maximizing the variance** of project data.

  SVM: maximizing the margin.

  Regression: minimizing the mean square errors (MSE)

  Sometimes the same model but different object function can lead different results. (Linear regression and ridge regression)

Introduction

1. Regression

   MSE, MAE, Huber Loss

2. Classification

   Cross entropy, Focal loss

3. Triple loss

Residual

- Residual: predicted value v.s. target value.

Regression:

$$y - \hat{y}$$

Classification (error):

$$sign(\hat{y}, y) = \begin{cases} 1 & \hat{y} = \hat{y} \\ 0 & \hat{y} \neq \hat{y} \end{cases}$$

$$error\ rate = \frac{1}{n}\sum_{i=1}^{n} sign(\widehat{y_i}, y_i)$$

MSE & MAE

Mean Square Error (MSE)

Mean Absolute Error (MAE)

<u>Why square or absolute?</u>

Target value: $y_1 = 0$, $y_2 = 1$

Predicted value: $\hat{y}_1 = 100$，$\hat{y}_2 = 99$

Residual 1=$y_1 - \hat{y}_1 = 0 - 100 = -100$

Residual 2=$y_2 - \hat{y}_2 = 1 - (-99) = 100$

Residual 1+ Residual 2 $= -100 + 100 = 0$

MSE & MAE

Mean Square Error (MSE)

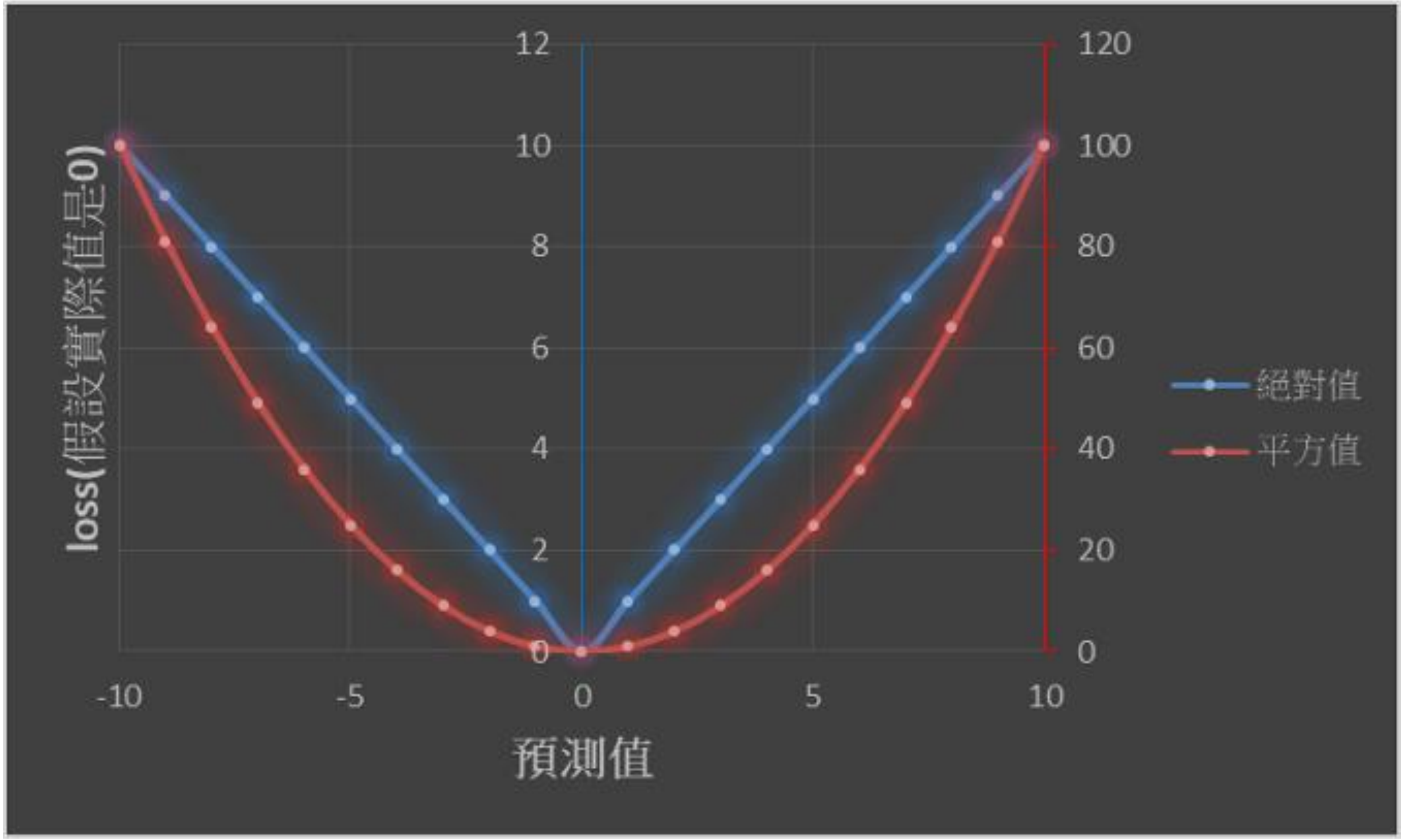
$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Mean Absolute Error (MAE)

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

MSE & MAE

Trend of residual (target value =0)



Comparison (MSE&MAE)

With a fair baseline, RMSE (root MSE)

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

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

Comparison (MSE&MAE)

ID	residual	| residual |	$residual^2$
1	-10	10	100
2	-5	5	25
3	0	0	0
4	5	5	25
5	10	10	100
MAE=6, RMSE=7.07			

Change ID5 with a outliner

ID	residual	| residual |	$residual^2$
1	-10	10	100
2	-5	5	25
3	0	0	0
4	5	5	25
5	100	100	10000
MAE=24, RMSE=45.06			

Problem of MSE: more outlier sensitivity.

Comparison (MSE&MAE)

- Problem of MAE: same gradient value.
- When loss is small, it's difficult to reach the optimal target.

$$f_1(x) = x^2, f_1'(x) = 2x$$

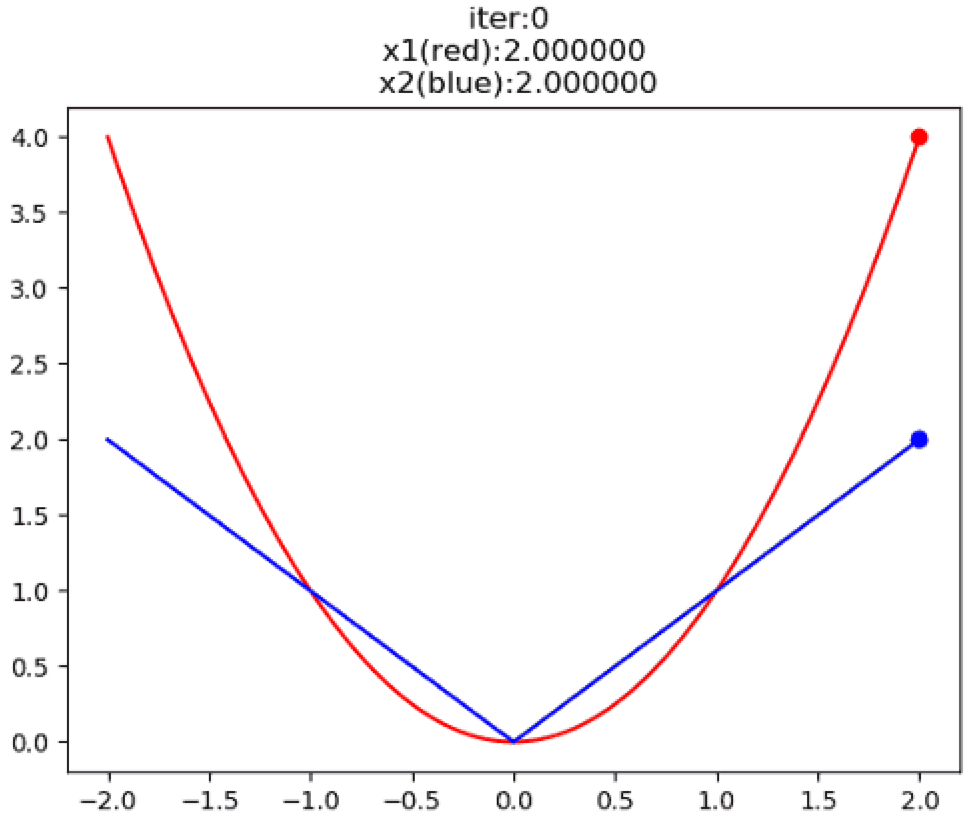
$$f_2(x) = |x|, f_2'(x) = \frac{x}{|x|}$$

Gradient update:

$$x^{t+1} \rightarrow x^t + rf'(x)$$

Suppose: $x^0 = 2, \ \ r = 0.3$

Comparison (MSE&MAE)



$f(x)=x^2$			
t	x^t	$f'(x)$	x^{t+1}
1	2	4	0.8
2	0.8	1.6	0.32
3	0.32	0.64	0.128
4	0.128	0.256	0.0512
5	0.0512	0.1024	0.02048
6	0.02048	0.04096	0.008192
7	0.008192	0.016384	0.003277
8	0.003277	0.006554	0.001311
9	0.001311	0.002621	0.000524
10	0.000524	0.001049	0.00021
11	0.000210	0.000419	0.000084
12	0.000084	0.000168	0.000034
13	0.000034	0.000067	0.000013
14	0.000013	0.000027	0.000005
15	0.000005	0.000011	0.000002
16	0.000002	0.000004	0.000001
17	0.000001	0.000002	0.000000
18	0.000000	0.000001	0.000000
19	0.000000	0.000000	0.000000
20	0.000000	0.000000	0.000000

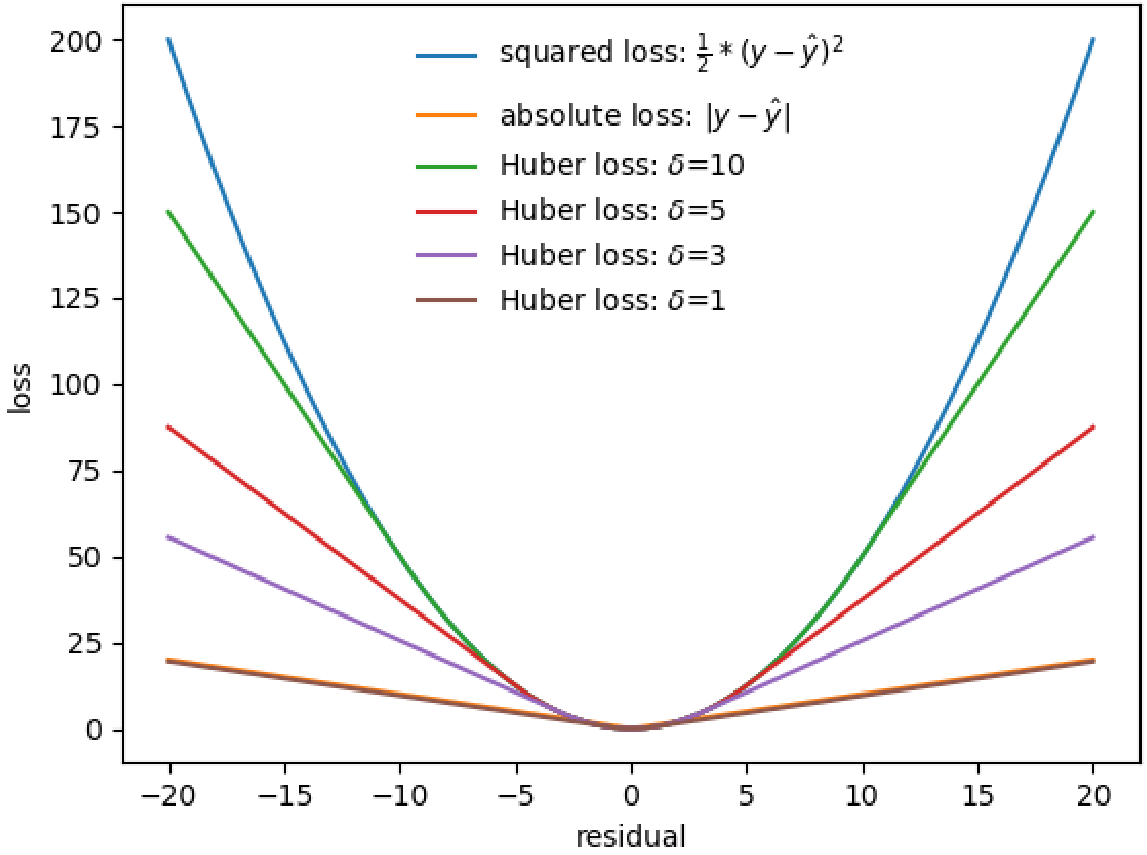
$f(x)=\|x\|$			
t	x^t	$f'(x)$	x^{t+1}
1	2	1	1.7
2	1.7	1	1.4
3	1.4	1	1.1
4	1.1	1	0.8
5	0.8	1	0.5
6	0.5	1	0.2
7	0.2	1	-0.1
8	-0.1	-1	0.2
9	0.2	1	-0.1
10	-0.1	-1	0.20
11	0.20	1	-0.1
12	-0.10	-1	0.20
13	0.20	1	-0.10
14	-0.10	-1	0.20
15	0.20	1	-0.10
16	-0.10	-1	0.20
17	0.20	1	-0.10
18	-0.10	-1	0.20
19	0.20	1	-0.10
20	-0.10	-1	0.20

Huber Loss

Huber loss:

$$Loss(y, \hat{y}) = \begin{cases} \frac{1}{2}(y-\hat{y})^2, & |y-\hat{y}| \le \delta \\ \delta(|y-\hat{y}| - \frac{1}{2}\delta), & O.W. \end{cases}$$

δ: parameter of Huber loss.



MAE, MSE & Huber Loss

ID	residual	| residual |	residual^2	Huber (δ =1)	Huber (δ =10)
1	-10	10	100	9.5	50
2	-5	5	25	4.5	12.5
3	0	0	0	0	0
4	5	5	25	4.5	12.5
5	10	10	100	9.5	50
MAE=6, RMSE=7.07 **MeanHuber(δ=1)=5.6, MeanHuber(δ=10)=25**					

ID	residual	| residual |	residual^2	Huber (δ =1)	Huber (δ =10)
1	-10	10	100	9.5	50
2	-5	5	25	4.5	12.5
3	0	0	0	0	0
4	5	5	25	4.5	12.5
5	100	100	10000	99.5	950
MAE=24, RMSE=45.06 **MeanHuber(δ=1)=23.6, MeanHuber(δ=10)=205**					

Classification

Classification:

$$sign(\hat{y}, y) = \begin{cases} 1 & y = \hat{y} \\ 0 & y \neq \hat{y} \end{cases}$$

$$error\ rate = 1 - \frac{1}{n}\sum_{i=1}^{n} sign(\widehat{y_i}, y_i)$$

We hope less error rate more better in classification.

Can we use the classification error rate/accuracy as loss function?

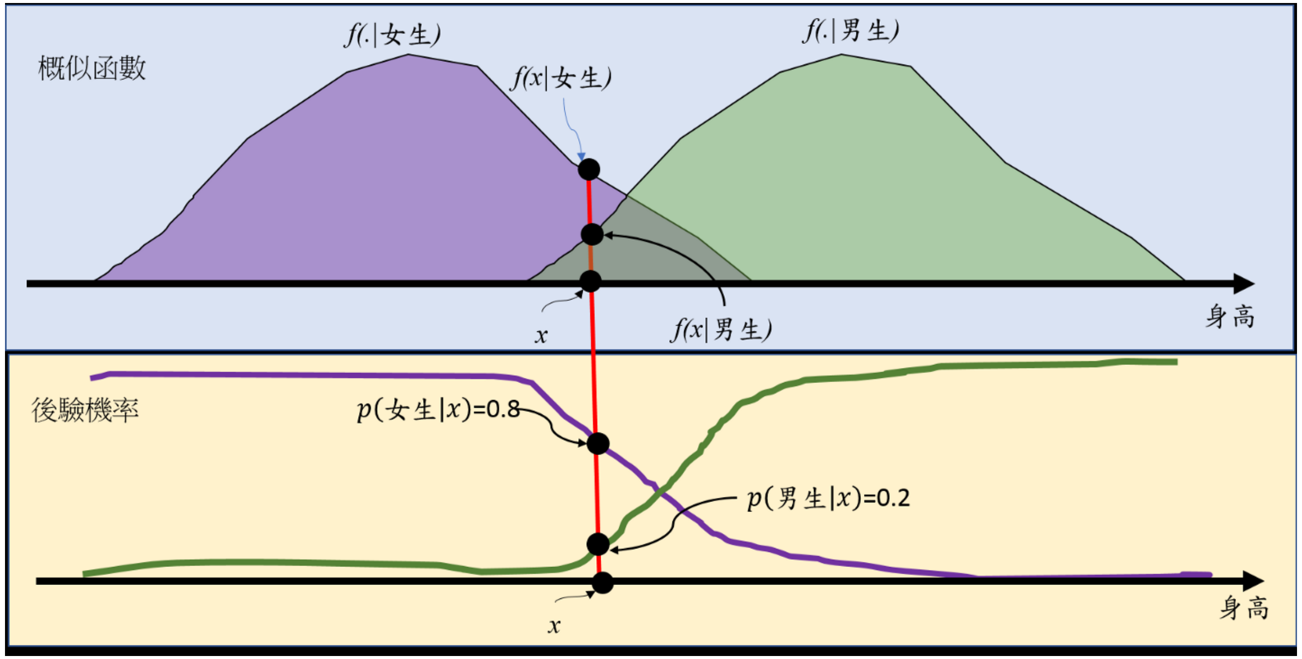
Classification

		Model 1 (輸出)				Model 2 (輸出)			
		機率輸出			判斷	機率輸出			判斷
	Target (Label)	男生	女生	其他		男生	女生	其他	
data 1	男生	0.4	0.3	0.3	男生 (正確)	0.7	0.1	0.2	男生 (正確)
data 2	女生	0.3	0.4	0.3	女生 (正確)	0.1	0.8	0.1	女生 (正確)
data 3	男生	0.5	0.2	0.3	男生 (正確)	0.9	0.1	0	男生 (正確)
data 4	其他	0.8	0.1	0.1	男生 (錯誤)	0.4	0.3	0.3	男生 (錯誤)
		模型1錯誤率: 1/4=0.25				模型2錯誤率: 1/4=0.25			

Can we observe any difference between model 1 & 2 from **error rate**?
NO…
BUT we can observe that model 2 has better probability outputs than model 1.
Error rate cannot as learning object for learning updating, it's just a metric for evaluating model performance.

Classification

- How do we make decision for a new sample in classification model?
- **ANS: posterior probability.**



Cross-entropy

- Cross-entropy is usually used in classification loss.
- **<u>Entropy</u>**: the average of information which is produced by a stochastic source of data.
- **<u>Information gain</u>**: (suppose X is a random variable)

$$I(x) = -log_2(p(x))$$

Information gain

A is stupid, and his grades usually are around 50 marks.

B is smart, and his grades usually are almost 100 marks.

Probability to pass the exam for A: $p(x_A) = 0.4$

$$I(x_A) = -log_2\big(p(x_A)\big) = 1.322$$

Probability to pass the exam for B: $p(x_B) = 0.99$

$$I(x_B) = -log_2\big(p(x_B)\big) = 0.014$$

Entropy

Entropy: the average of information which is produced by a stochastic source of data.

In information theory,

$$\text{Entropy} = \text{ Shannon entropy}$$

$$H(X) = \sum_{i} -p_i log_2(p_i)$$

Generally, entropy refers to uncertainty for the random variable X.

Entropy

$p(x_A = pass) = 0.4, \qquad p(x_A = fail) = 0.6$

$p(x_B = pass) = 0.99, \qquad p(x_B = fail) = 0.01$

$$H(X) = \sum_i -p_i log_2(p_i)$$

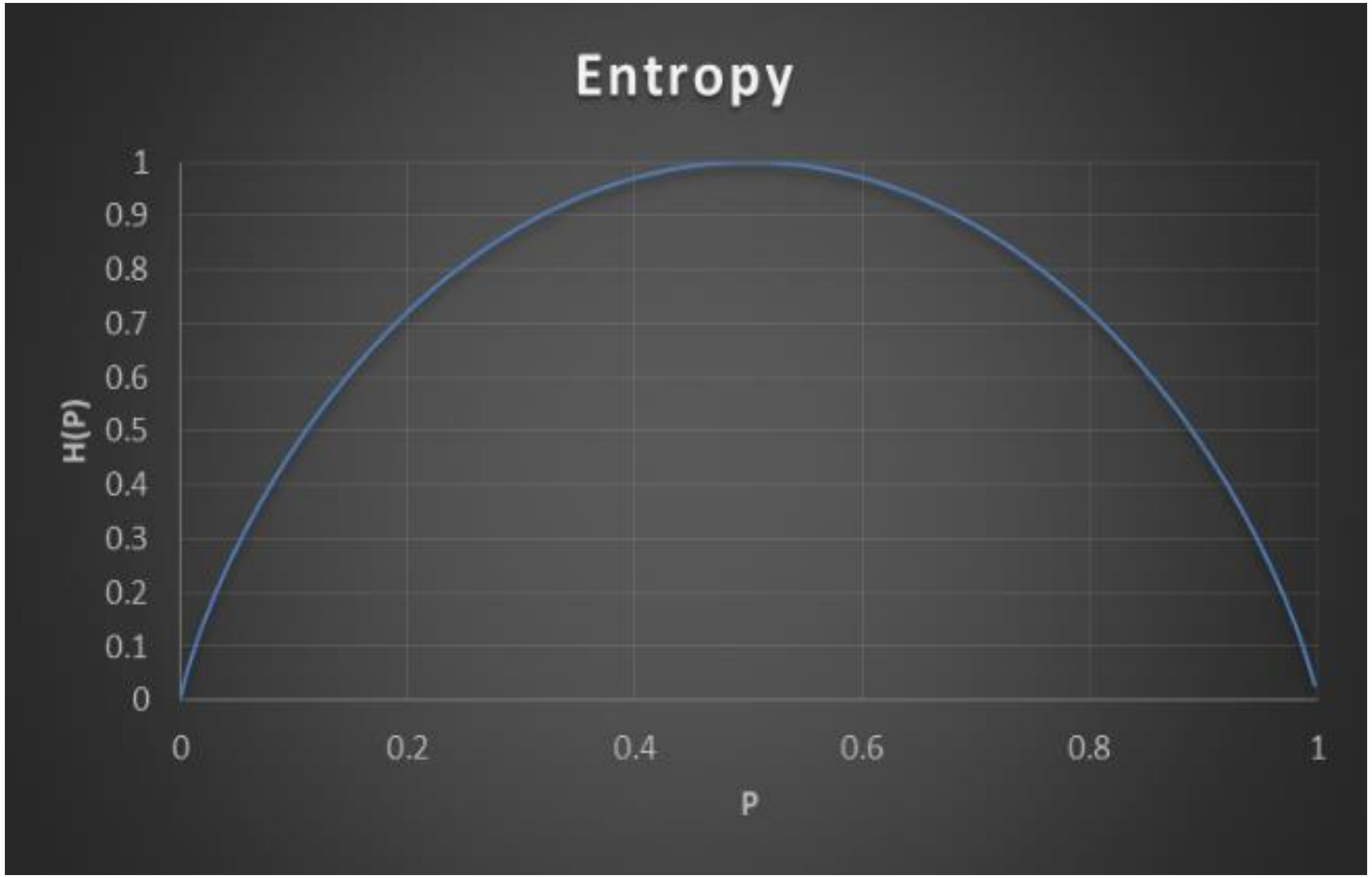
$$H(X_A) = -0.4\log(0.4) - 0.6\log(0.6) = 0.971$$
$$H(X_B) = -0.99\log(0.99) - 0.01\log(0.01) = 0.081$$

Same conclusion for information gain.

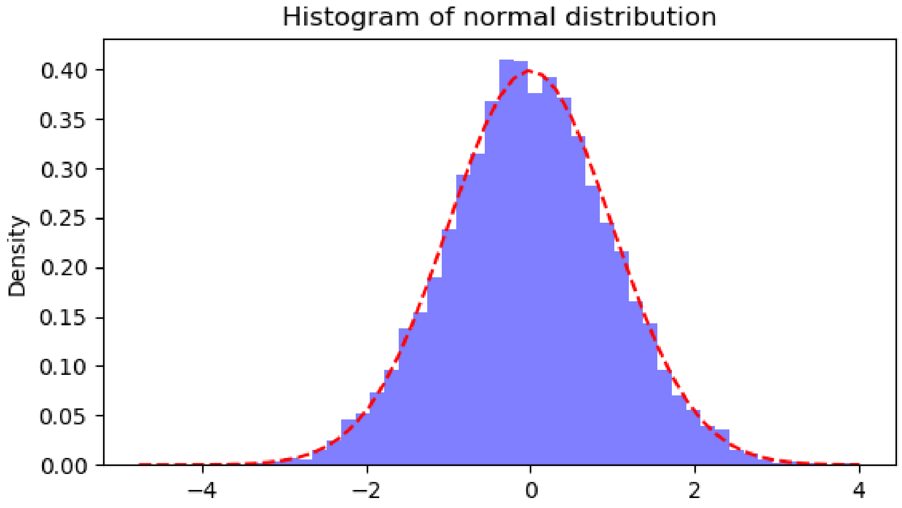
$$I(x_A) = -log_2\big(p(x_A)\big) = 1.322$$
$$I(x_B) = -log_2\big(p(x_B)\big) = 0.014$$

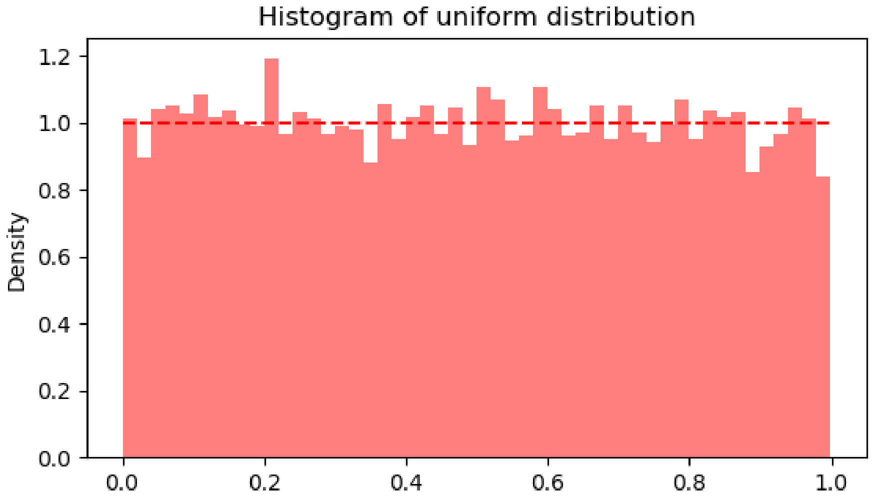
Entropy

When p=0.5 has the largest entropy.



Entropy





$p(X_A) = \begin{cases} 0.1 & x = 1 \\ 0.15 & x = 2 \\ 0.5 & x = 3 \\ 0.15 & x = 4 \\ 0.1 & x = 5 \end{cases}$	$p(X_A) = \begin{cases} 0.01 & x = 1 \\ 0.09 & x = 2 \\ 0.8 & x = 3 \\ 0.09 & x = 4 \\ 0.01 & x = 5 \end{cases}$	$p(X_B) = \begin{cases} 0.2 & x = 1 \\ 0.2 & x = 2 \\ 0.2 & x = 3 \\ 0.2 & x = 4 \\ 0.2 & x = 5 \end{cases}$
$H(X_A) = 1.985$	$H(X_A) = 1.016$	$H(X_B) = 2.322$

Cross-entropy

Formula of cross- entropy:

$$H = \sum_{i=1}^{n} \sum_{c=1}^{C} -y_{c,i} log_2(p_{c,i})$$

C: number of class (male, female, other)

n: number of data

$y_{c,i}$: binary indicator (0 or 1) from one hot encode (i-th data assigns to c-class)

$p_{c,i}$: probability of i-th data assigns to c-class

One hot encode
(Dummy variable)

Data 1	Male
Data 2	Female
Data 3	Male
Data 4	Other

Male
Female
Other

	Male	Female	Other
Data 1	1	0	0
Data 2	0	1	0
Data 3	1	0	0
Data 4	0	0	1

Cross-entropy

$$H = \sum_{i=1}^{n} \sum_{c=1}^{C} -y_{c,i} log_2(p_{c,i})$$

		Model 1 (輸出)					
		機率輸出			實際One-hot encode		
	Target (Label)	男生	女生	其他	男生	女生	其他
data 1	男生	0.4	0.3	0.3	1	0	0
data 2	女生	0.3	0.4	0.3	0	1	0
data 3	男生	0.5	0.2	0.3	1	0	0
data 4	其他	0.8	0.1	0.1	0	0	1
		模型1錯誤率: 1/4=0.25 Cross-entropy=6.966					

Data 1:

$$\sum_{c=1}^{C} -y_{c,1} log_2(p_{c,1})$$
$$= -1 * \log(0.4) - 0 * \log(0.3) - 0 * \log(0.3) = 1.322$$

Data 4:

$$\sum_{c=1}^{C} -y_{c,4} log_2(p_{c,4})$$
$$= -0 * \log(0.8) - 0 * \log(0.1) - 1 * \log(0.1) = 3.3219$$

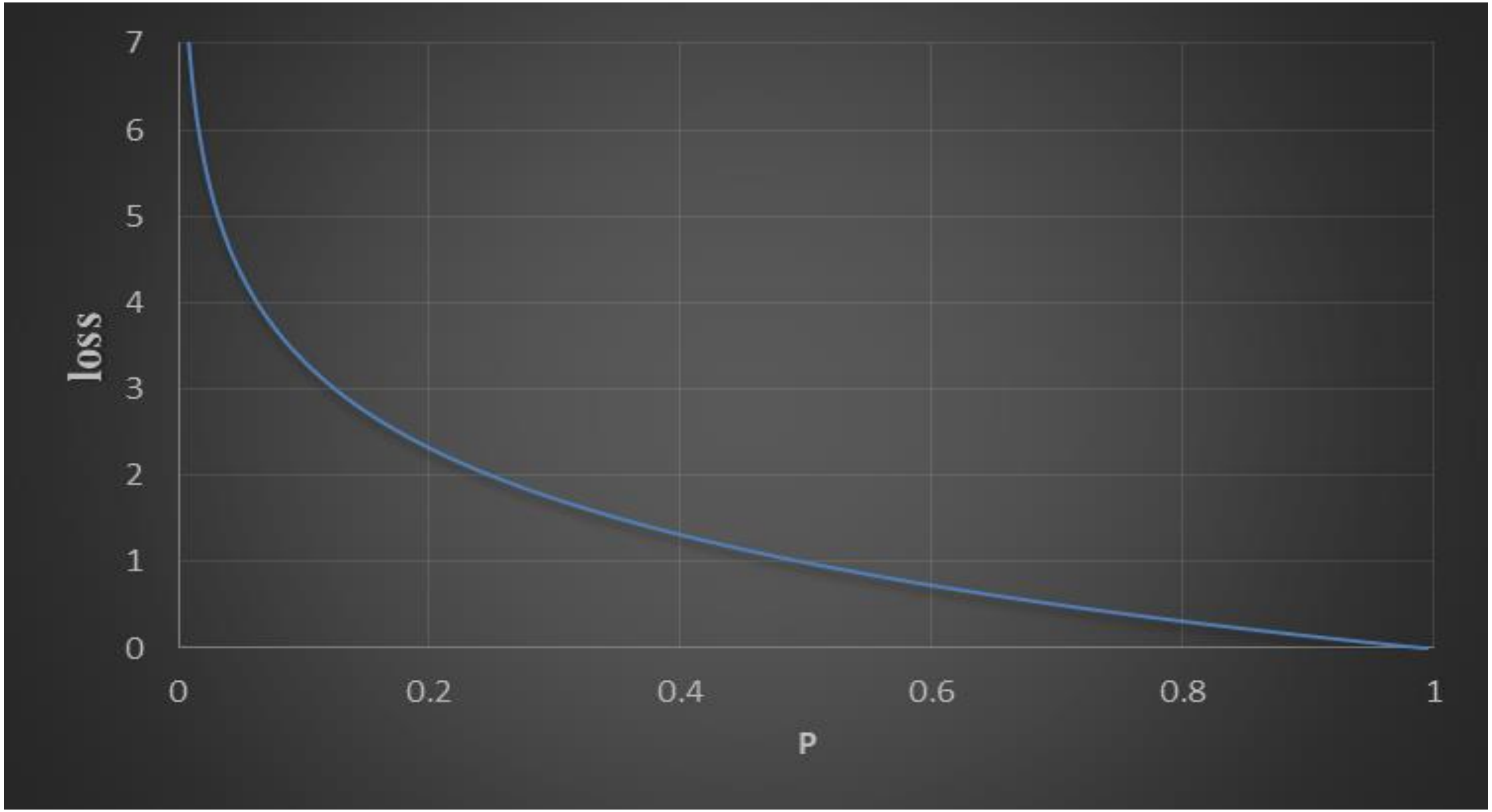
SO less probability data has larger loss function (entropy value)→learning target.

Cross-entropy
for evaluating the model performance

		Model 1 (輸出)					
		機率輸出			實際One-hot encode		
	Target (Label)	男生	女生	其他	男生	女生	其他
data 1	男生	0.4	0.3	0.3	1	0	0
data 2	女生	0.3	0.4	0.3	0	1	0
data 3	男生	0.5	0.2	0.3	1	0	0
data 4	其他	0.8	0.1	0.1	0	0	1
		模型1錯誤率: 1/4=0.25 Cross-entropy=6.966					

		Model 2 (輸出)					
		機率輸出			實際One-hot encode		
	Target (Label)	男生	女生	其他	男生	女生	其他
data 1	男生	0.7	0.1	0.2	1	0	0
data 2	女生	0.1	0.8	0.1	0	1	0
data 3	男生	0.9	0.1	0	1	0	0
data 4	其他	0.4	0.3	0.3	0	0	1
		模型1錯誤率: 1/4=0.25 Cross-entropy= 2.310					

Cross-entropy for loss function

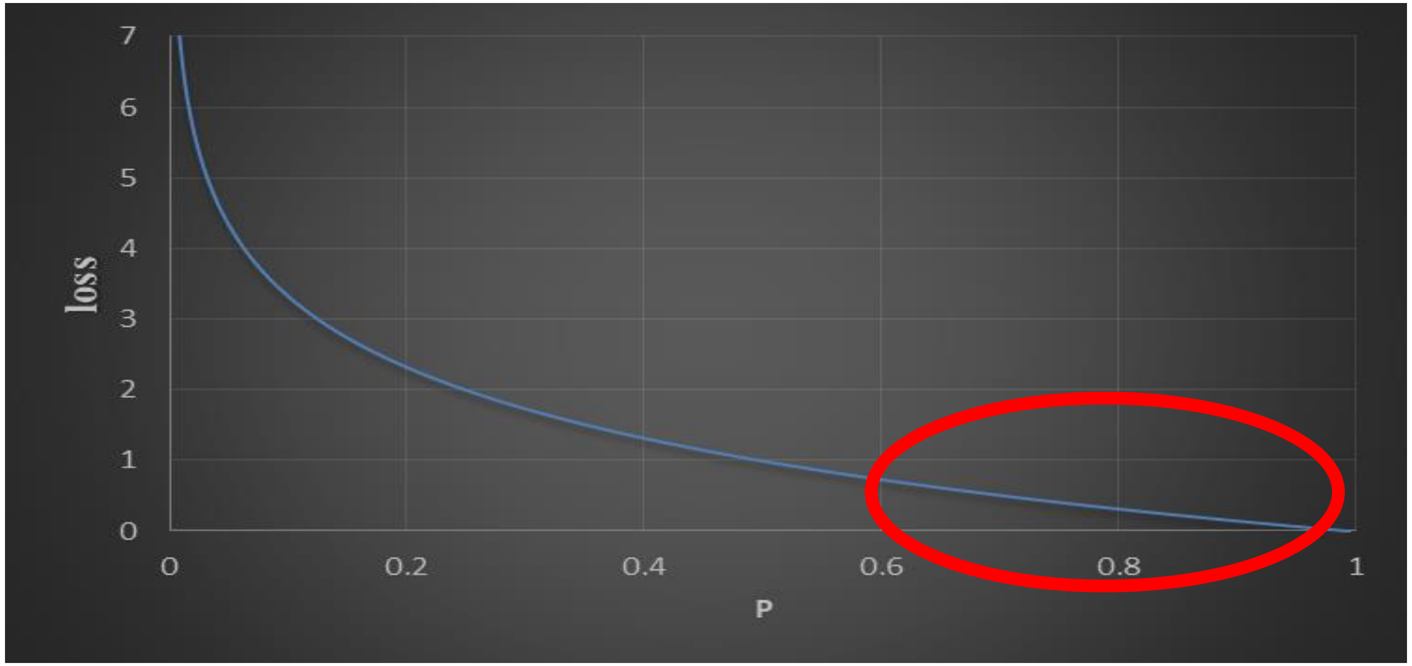


Focal loss (1/3)

Cross-entropy (CE) for $y \in \{\pm 1\}$

$$CE(p, y) = \begin{cases} -\log(p), & if\ y = 1 \\ -\log(1-p), & if\ y = -1 \end{cases}$$

$$CE(p, y) = CE(p_t) = -\log(p_t), \qquad p_t = \begin{cases} p, & if\ y = 1 \\ 1-p, & if\ y = -1 \end{cases}$$



Focal loss (2/3)

α-balanced cross-entropy:

$$CE(p_t) = -\alpha \log(p_t)$$

BUT it's not effect for larger class unbalance problem.

Modulating factor:

$$(1 - p_t)^r$$

r: focusing parameter, $r \geqq 0$.

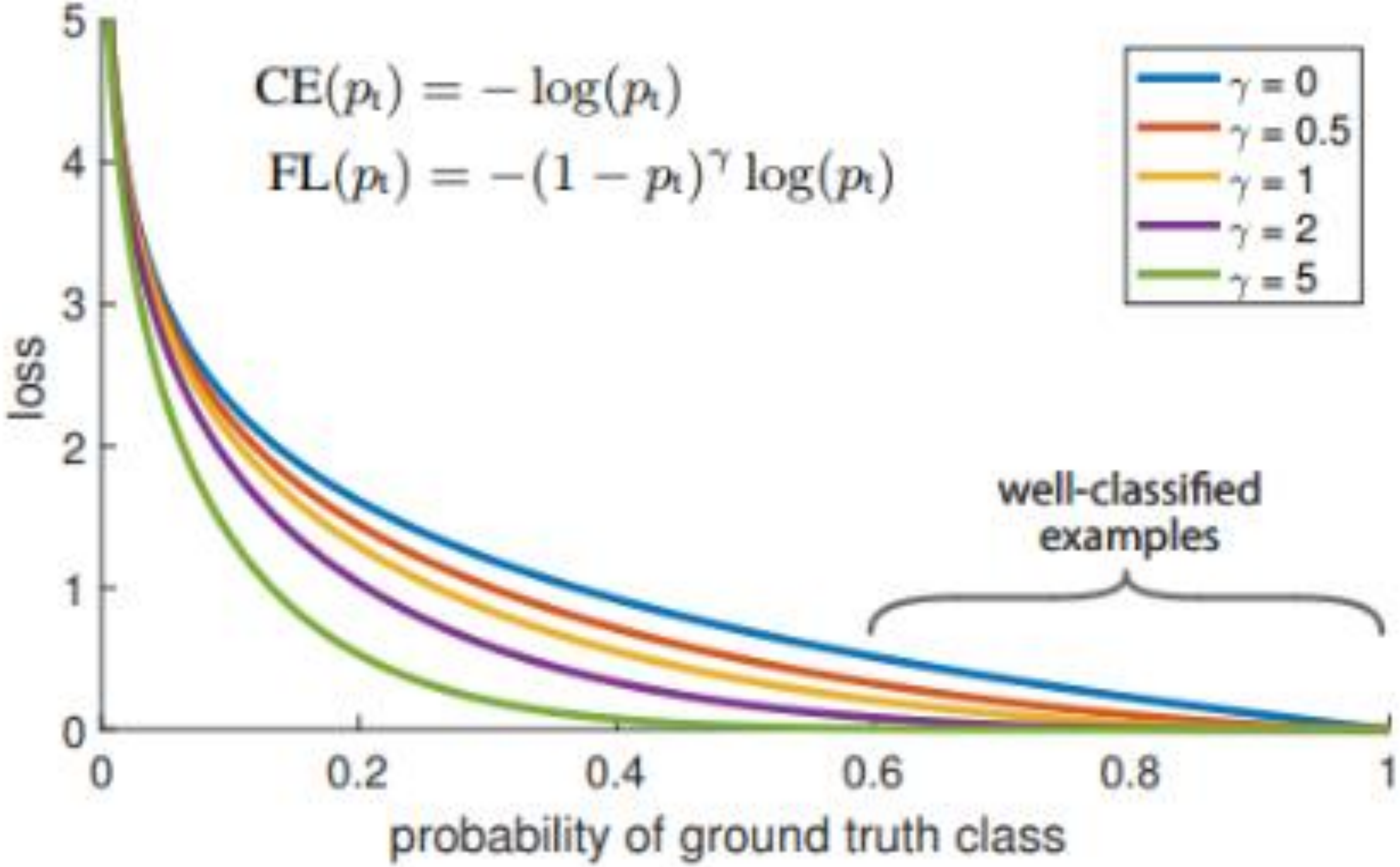
Focal loss:

$$FL(p_t) = -(1 - p_t)^r \log(p_t)$$

α-balanced focal loss:

$$FL(p_t) = -\alpha(1 - p_t)^r \log(p_t)$$

Focal loss (3/3)



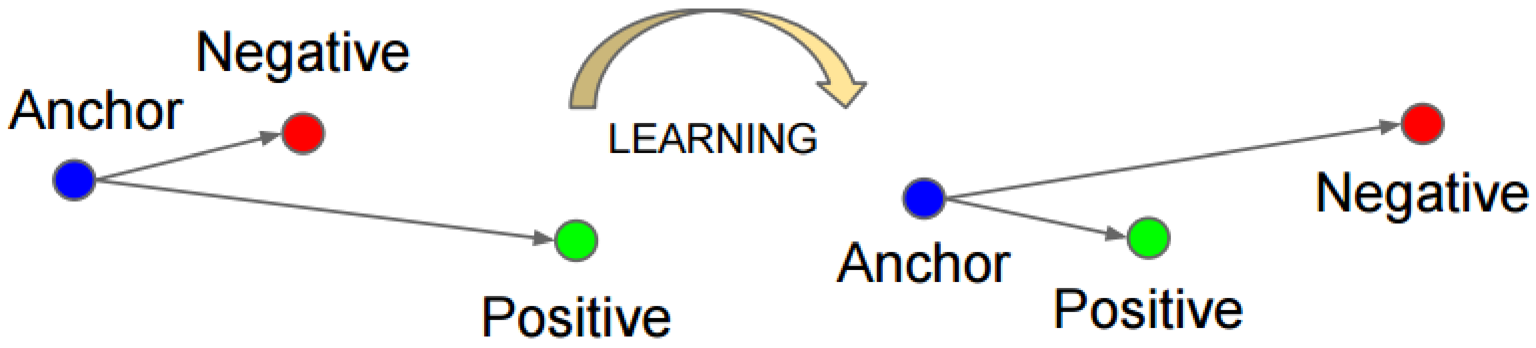
Triple Loss

Triple

Anchor: a randomly training data with label c

Positive: training data in label c

Negative: training data in other labels



Triple Loss

$\mathit{Anchor}: x_i^a$

$\mathit{Positive}: x_i^p$

$\mathit{Negative}: x_i^n$

$\text{Encoder network}: f(x)$

$\mathit{Anchor}: f(x_i^a)$

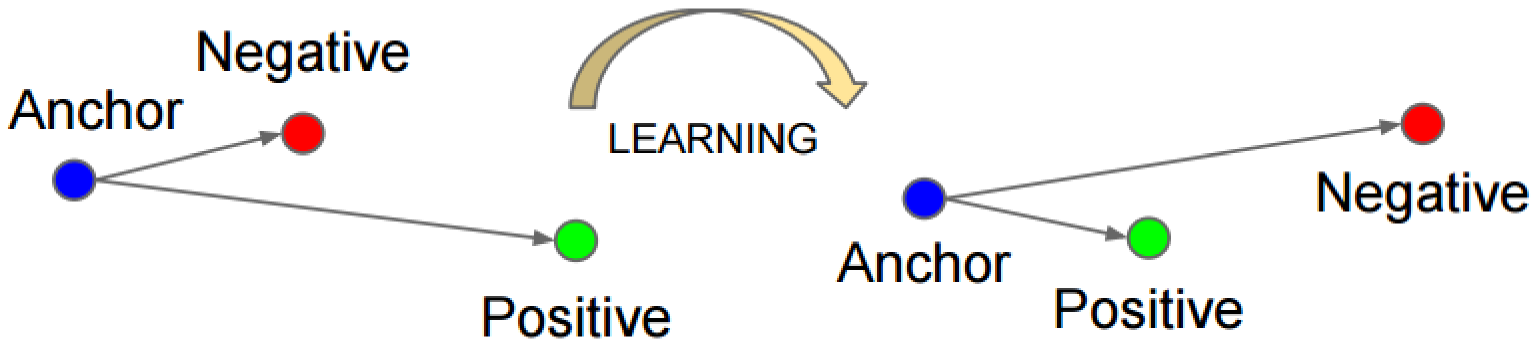
$\mathit{Positive}: f(x_i^p)$

$\mathit{Negative}: f(x_i^n)$

triple loss aims to

$dist(f(x_i^a),(x_i^p))\downarrow$

$dist(f(x_i^a),(x_i^n))\uparrow$



Triple Loss

$$dist(f(x_i^a), f(x_i^p)) + \alpha < dist(f(x_i^a), f(x_i^n))$$

$$\Rightarrow \left\| f(x_i^a) - f(x_i^p) \right\|_2^2 + \alpha < \left\| f(x_i^a) - f(x_i^n) \right\|_2^2$$

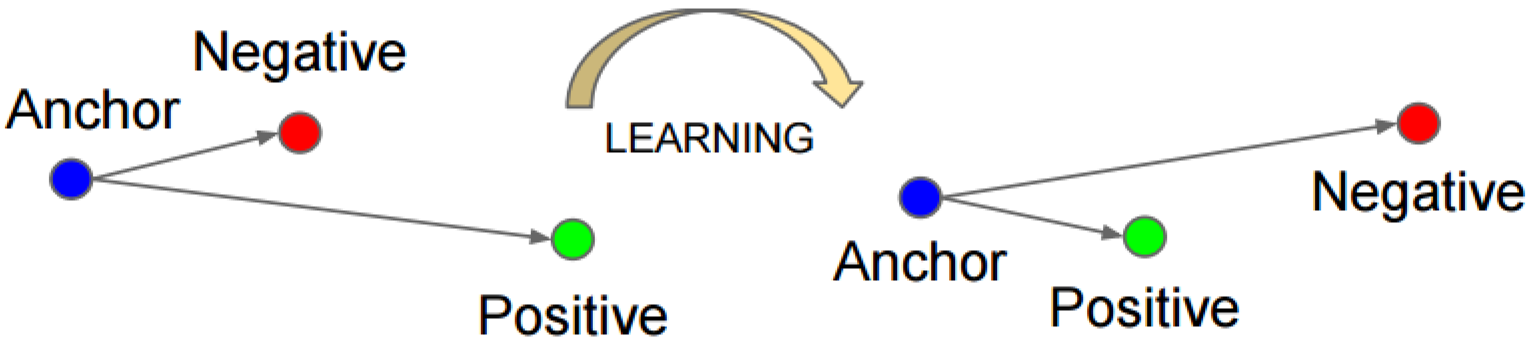
$$\arg\min\{\sum_i \left(\left\| f(x_i^a) - f(x_i^p) \right\|_2^2 + \alpha - \left\| f(x_i^a) - f(x_i^n) \right\|_2^2 \right)\}$$

$$\Downarrow$$

$$\arg\min \sum_i \left[\left\| f(x_i^a) - f(x_i^p) \right\|_2^2 + \alpha - \left\| f(x_i^a) - f(x_i^n) \right\|_2^2 \right]_+$$

Triple Loss

$$\left[d_p - d_n + \alpha\right]_+ = \begin{cases} d_p - d_n + \alpha & d_p - d_n + \alpha > 0 \\ 0 & d_p - d_n + \alpha < 0 \end{cases}$$

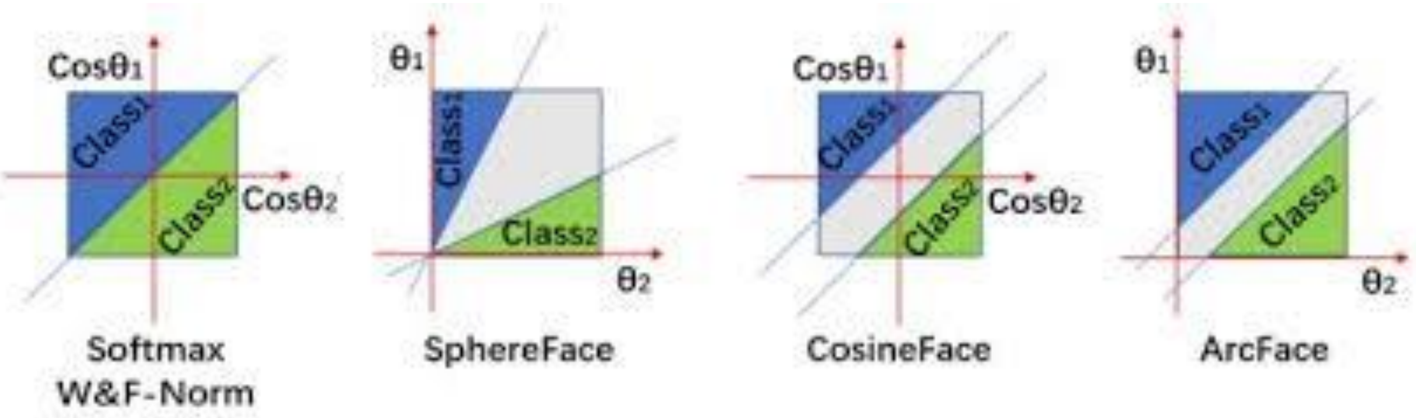


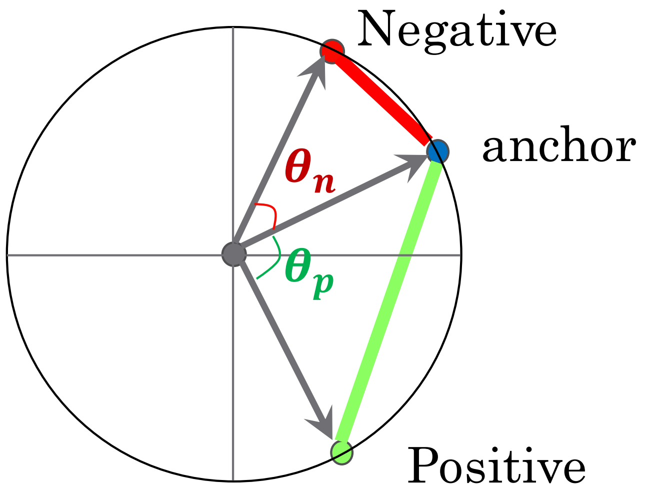
Conclusion

MSE, MAE, Huber loss, triple loss do the same thing.

Similarity measurement.

Cosine loss





Can MSE be a loss for classification?