

Loss fn

L₁ Vs L₂

L₂ Vs Cross entropy

Cross entropy Vs Softmax Loss

Softmax Vs L-Softmax

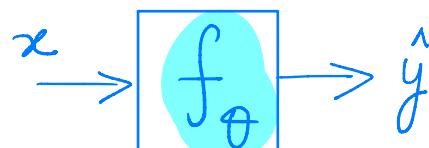
L-Softmax Vs A-Softmax

A Softmax Vs AM-Softmax.

Supervised Learning



Features $\rightarrow x$
 Targets $\rightarrow y$
 Predictions $\rightarrow \hat{y}$
 Parameters $\rightarrow \theta$



$$y = f(x)$$

we are interested to somehow predicting our θ parameters.

Loss fn \Rightarrow Say $J(\theta)$ is function of (y, \hat{y})

$\therefore \theta \rightarrow$ Randomly initialized

$$\hat{\theta} = \arg \min_{\theta} J(\theta)$$

$$\theta^{\text{new}} = \theta^{\text{old}} - \eta \frac{\partial J(\theta)}{\partial \theta}$$



Gradient descent

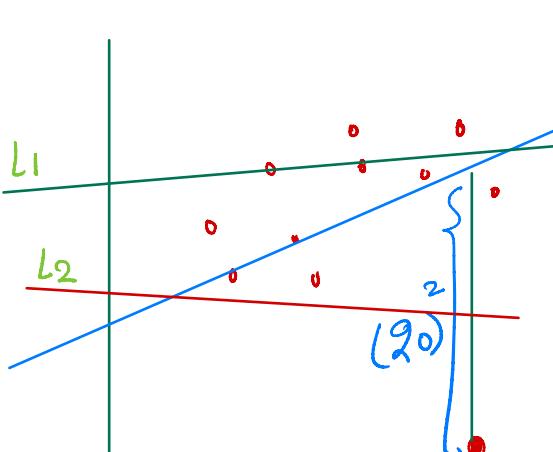
- Samplewise
- full data
- Batchwise

SGD

we need to choose the direction that will decrease $J(\theta)$

Loss fn or Cost fn choices can be

$$L_2 : \frac{1}{2} (y - \hat{y})^2 \quad (\text{MSE})$$



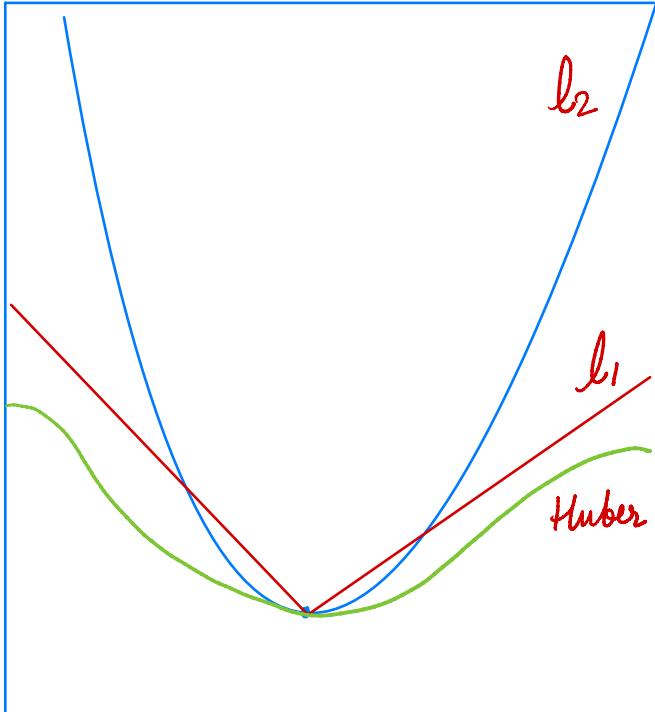
original
(Sensitivity)

MSE heavily penalizes

outliers. This causes outliers to manipulate the learning heavily biased towards them

$$L_1 : |y - \hat{y}| \quad (\text{MAE})$$

[Also called as Manhattan distance.]



Huber loss takes the best of both
 Near to origin \Rightarrow Close to L_2
 away from origin \Rightarrow Close to L_1

L_2 loss function Vs Cross Entropy

* found truth $\Rightarrow y$ and prediction $\Rightarrow \hat{y}$

* total data points (m) we will consider $m=1$

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

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

(Regression)

(Naturally Coming for
linear regression)

* for classification MSE don't penalizes much the misclassification as compared to (CE).

Binary Cross Entropy

$$= -\frac{1}{m} \sum [y_i \ln(\hat{y}_i) + (1-y_i) \ln(1-\hat{y}_i)]$$

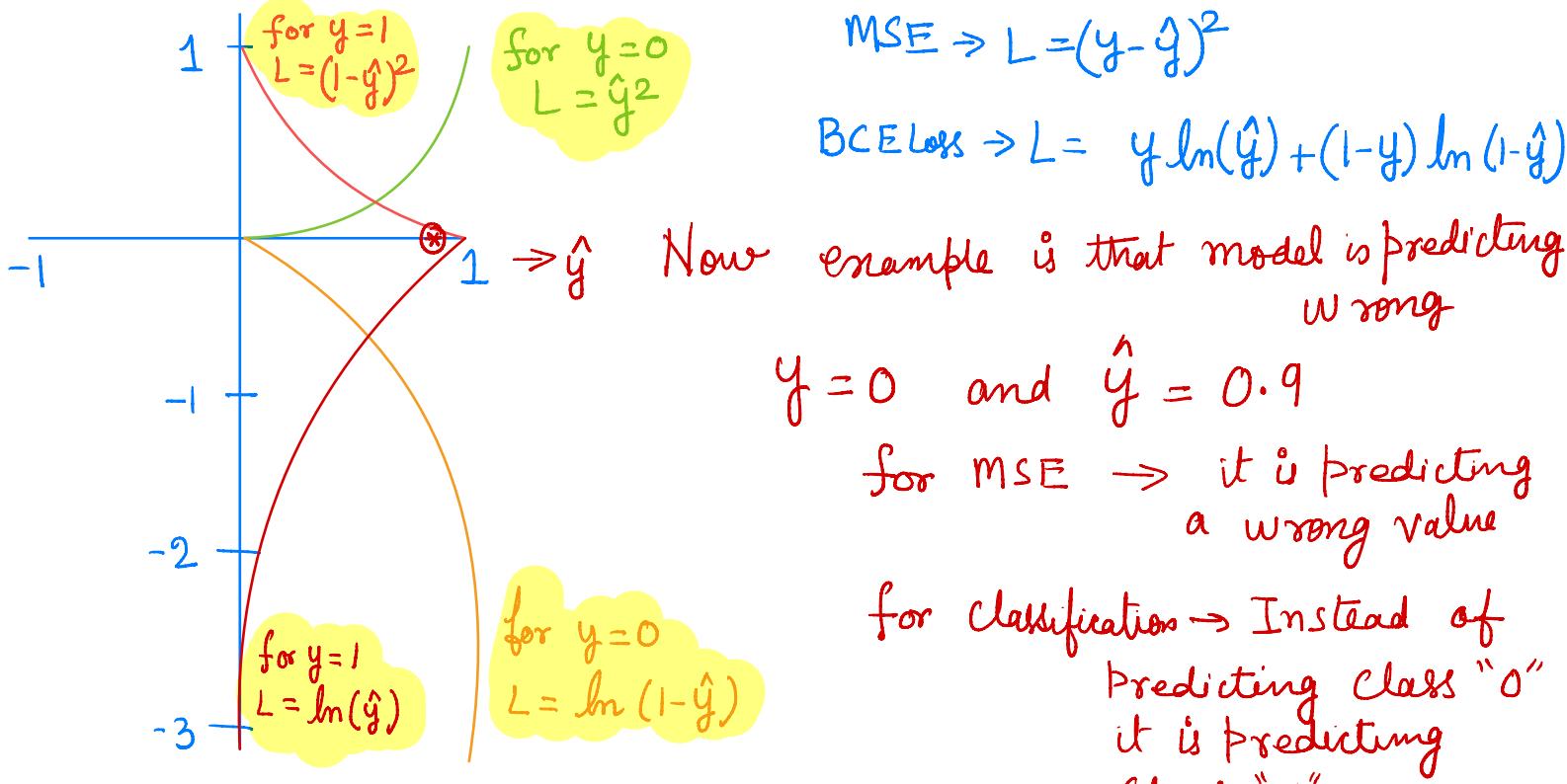
$$[y \ln(\hat{y}) + (1-y) \ln(1-\hat{y})]$$

Classification

(-ve sign is omitted as we are only interested in the shape of curve)

(Naturally Coming for logistic regression)

$$\frac{1}{1+e^x}$$



$$\frac{\partial L_{\text{MSE}}}{\partial \hat{y}} = 1.8$$

$$\frac{\partial L_{\text{CE}}}{\partial \hat{y}} = 10.00$$

$$\frac{\partial L}{\partial w} = \frac{\partial L}{\partial \hat{y}} * \frac{\partial \hat{y}}{\partial w}$$

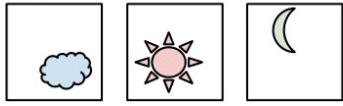
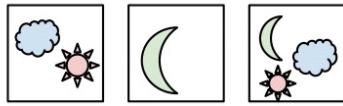
Then which loss function can change/update the weights effectively in the case of mis-classification

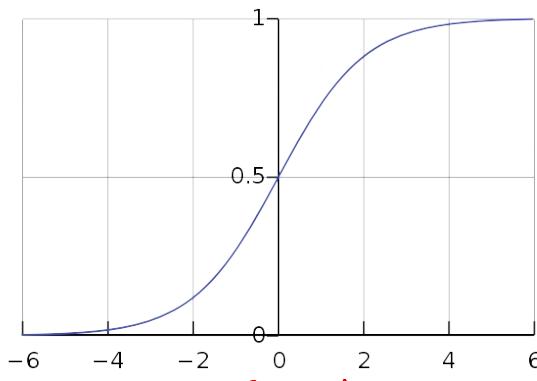
Regression : Continuous Variable

Small error may lead to small update. Update scale of both of them is different.

Classification : Discrete Variable

Small variation basically may lead to some other class
 (we are dealing with classes/categories)

	Multi-Class	Multi-Label
C = 3	<p>Samples</p>  <p>Labels (t)</p> <p>[0 0 1] [1 0 0] [0 1 0]</p>	<p>Samples</p>  <p>Labels (t)</p> <p>[1 0 1] [0 1 0] [1 1 1]</p>



$$f(s_i) = \frac{1}{1 + e^{-s_i}}$$

$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}}$$

Sigmoid is squashing function [logistic fn]

Softmax fn is not a loss function

It also squashes things b/w [0 & 1], adding upto 1.

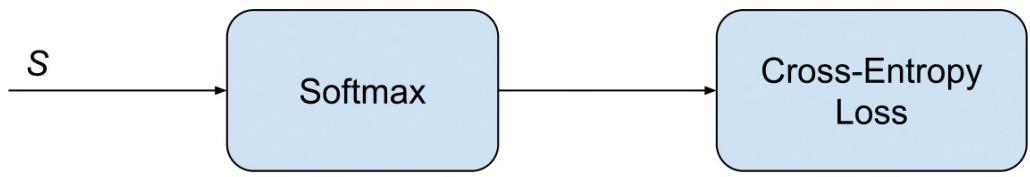
$$CE = - \sum_i^C t_i \log(s_i)$$

$$CE = - \sum_{i=1}^{C'=2} t_i \log(s_i) = -t_1 \log(s_1) - (1 - t_1) \log(1 - s_1)$$

CE loss

Binary Cross Entropy

Activation function (Sigmoid/Softmax)
is applied before CE/BCE losses.



$$f(s)_i = \frac{e^{s_i}}{\sum_j^C e^{s_j}} \quad CE = -\sum_i^C t_i \log(f(s)_i)$$

$$CE = -\log \left(\frac{e^{s_p}}{\sum_j^C e^{s_j}} \right)$$

Here
 (S_p)
 is the
 (Score for the +ve
 class)

$$\frac{\partial}{\partial s_p} \left(-\log \left(\frac{e^{s_p}}{\sum_j^C e^{s_j}} \right) \right) = \left(\frac{e^{s_p}}{\sum_j^C e^{s_j}} - 1 \right)$$

The derivative of CE wrt (S_p) & (S_n)

$$\frac{\partial}{\partial s_n} \left(-\log \left(\frac{e^{s_p}}{\sum_j^C e^{s_j}} \right) \right) = \left(\frac{e^{s_n}}{\sum_j^C e^{s_j}} \right)$$

Focal loss: It is a Cross-

Entropy loss that weights the contribution of each sample to the loss, based on its own classification error.

① Sample Correctly classified

then its weight is decreased.

② Mis-classified \Rightarrow Weight is increased.

$$FL = - \sum_{i=1}^{C=2} (1 - s_i)^\gamma t_i \log(s_i)$$

focusing factor *Ground truth*
Score
($r \geq 0$) focusing parameter

$(r=0)$ BCE

$$FL = \begin{cases} -(1 - s_1)^\gamma \log(s_1) & \text{if } t_1 = 1 \\ -(1 - (1 - s_1))^\gamma \log(1 - s_1) & \text{if } t_1 = 0 \end{cases}$$

$$FL = (1 - s_1)^\gamma t_1 \log(s_1) + (1 - s_2)^\gamma t_2 \log(s_2)$$

$\left. \begin{array}{l} \\ \\ \end{array} \right\} \begin{array}{l} s_2 \Rightarrow (1 - s_1) \\ t_1 = 0 \end{array}$

$$(1 - s_1)^\gamma \log(s_1)$$

$$(1 - s_2)^\gamma \log(s_2)$$

$$(1 - (1 - s_1))^\gamma \log(1 - s_1)$$

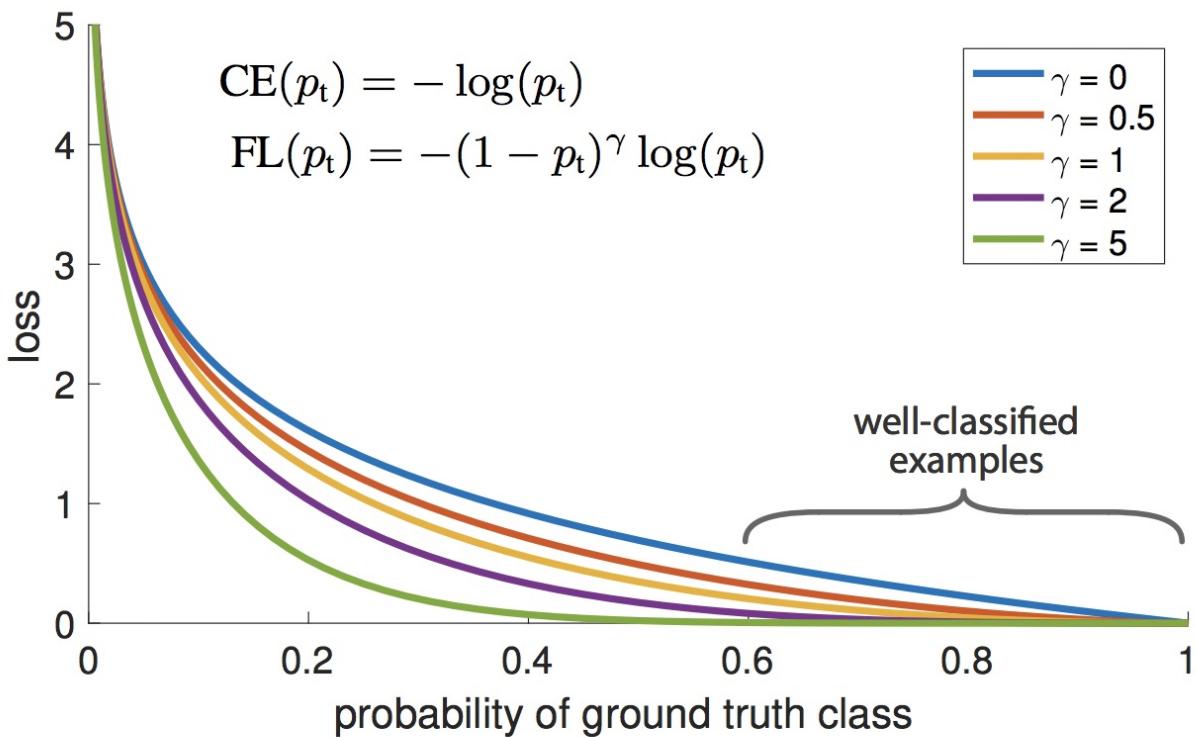
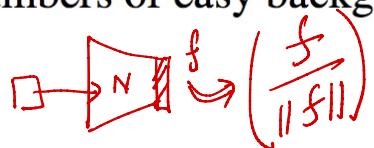
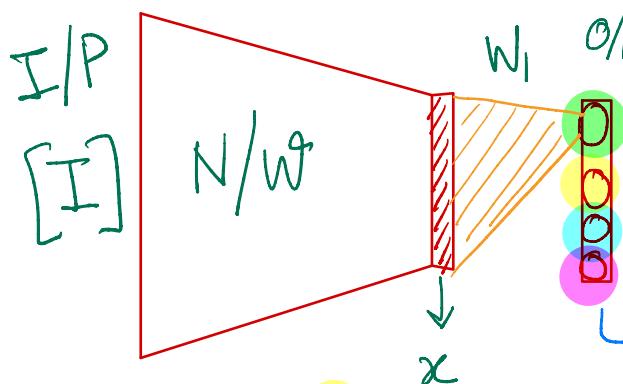


Figure 1. We propose a novel loss we term the *Focal Loss* that adds a factor $(1 - p_t)^\gamma$ to the standard cross entropy criterion. Setting $\gamma > 0$ reduces the relative loss for well-classified examples ($p_t > .5$), putting more focus on hard, misclassified examples. As our experiments will demonstrate, the proposed focal loss enables training highly accurate dense object detectors in the presence of vast numbers of easy background examples.

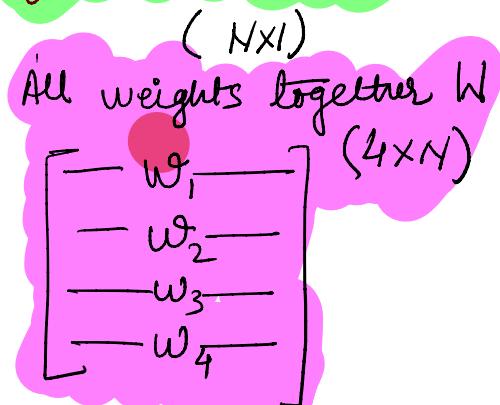


* Just talk about Triplet loss and its variants
 ⇒ Moving towards metric learning. [old slides]



Preactivated O/P
of our N/w
(N-dim) $[N \times 1]$
 $x: [x_1 \ x_2 \ \dots \ x_N] \in \mathbb{R}^N$

Any (w_i) is the weight vector of that $(i)^{\text{th}}$ neuron for that $(i)^{\text{th}}$ class.

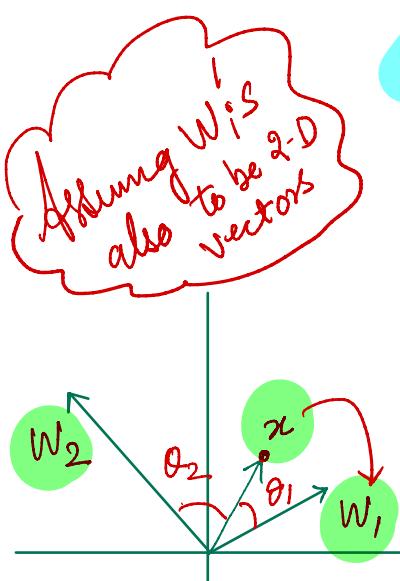


y' (Raw value)
 y' is the O/P
of the N/w we
get (4×1)

$$(w/o \text{ Softmax}) \quad \hat{y} = [y'_1 \ y'_2 \ y'_3 \ y'_4]$$

↓ Apply softmax
 $\hat{y} \Rightarrow$ after Softmax

$$y'_i = w_i \cdot x$$



$y \Rightarrow$ 1-Hot encoded
ground truth
 (4×1)

$$\hat{y} = [\hat{y}_1, \hat{y}_2, \hat{y}_3, \hat{y}_4]$$

$$\hat{y}_i = \frac{e^{w_i \cdot x}}{\sum_{j=1}^c e^{w_j \cdot x}}$$

Considering binary classification

$$\hat{y}_1 = \frac{e^{w_1 \cdot x}}{e^{w_1 \cdot x} + e^{w_2 \cdot x}} \quad \hat{y}_2 = \frac{e^{w_2 \cdot x}}{e^{w_1 \cdot x} + e^{w_2 \cdot x}}$$

$$w_1 \cdot \cos \theta_1 > w_2 \cdot \cos \theta_2$$

$$\text{Class 1} \stackrel{\circ}{\circ} \quad \hat{y}_1 > \hat{y}_2$$

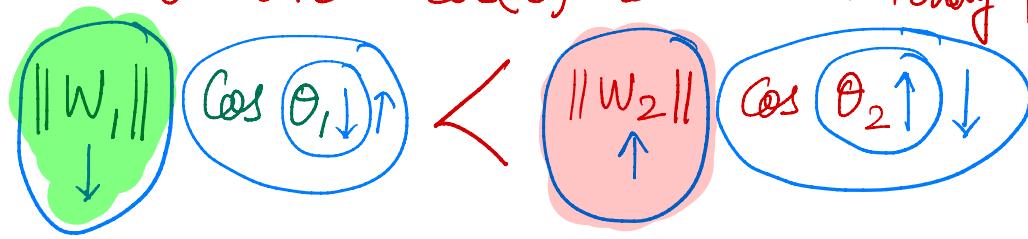
$$w_2 \cdot \cos \theta_2 > w_1 \cdot \cos \theta_1$$

$$\text{Class 2} \stackrel{\circ}{\circ} \quad \hat{y}_2 > \hat{y}_1$$

angle wise \textcircled{x} need to go to (w_1) but will go to $\textcircled{w_2}$!!!

Classification
boundary

* θ and $\cos(\theta)$ both are inversely proportional

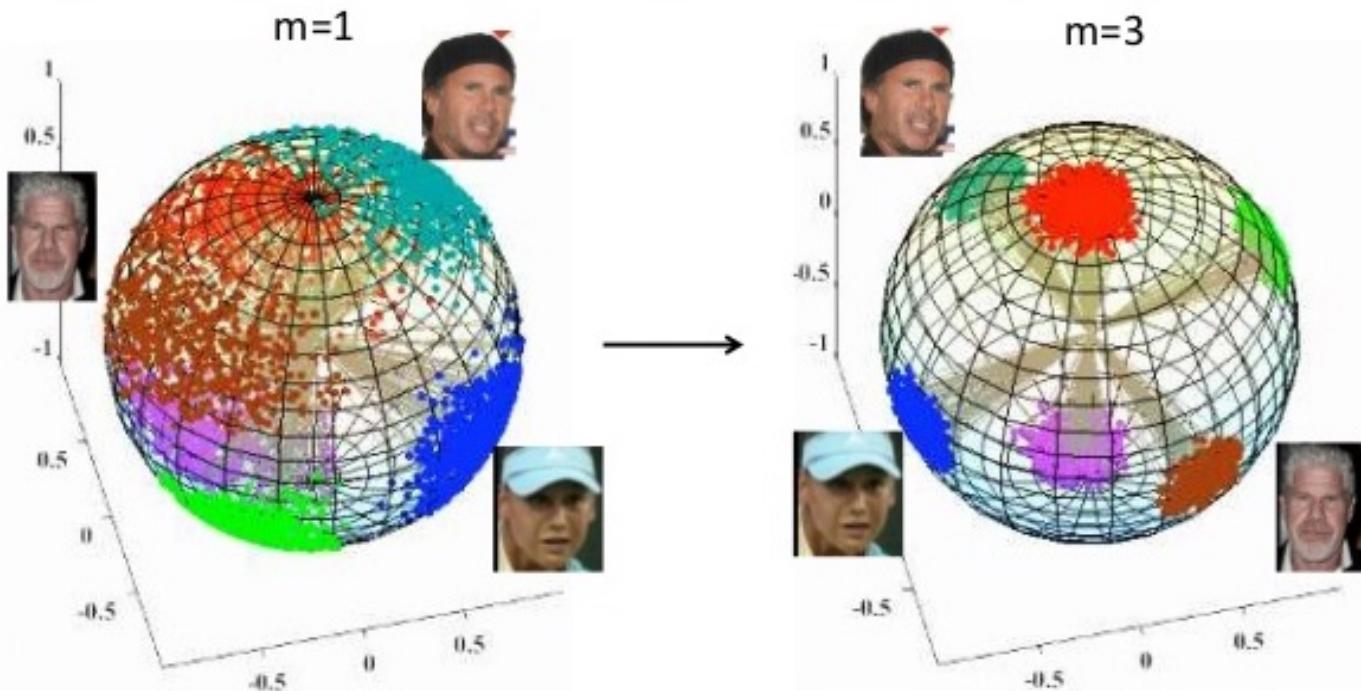


for Softmax loss function the decision boundary depends upon $\|W_i\|$

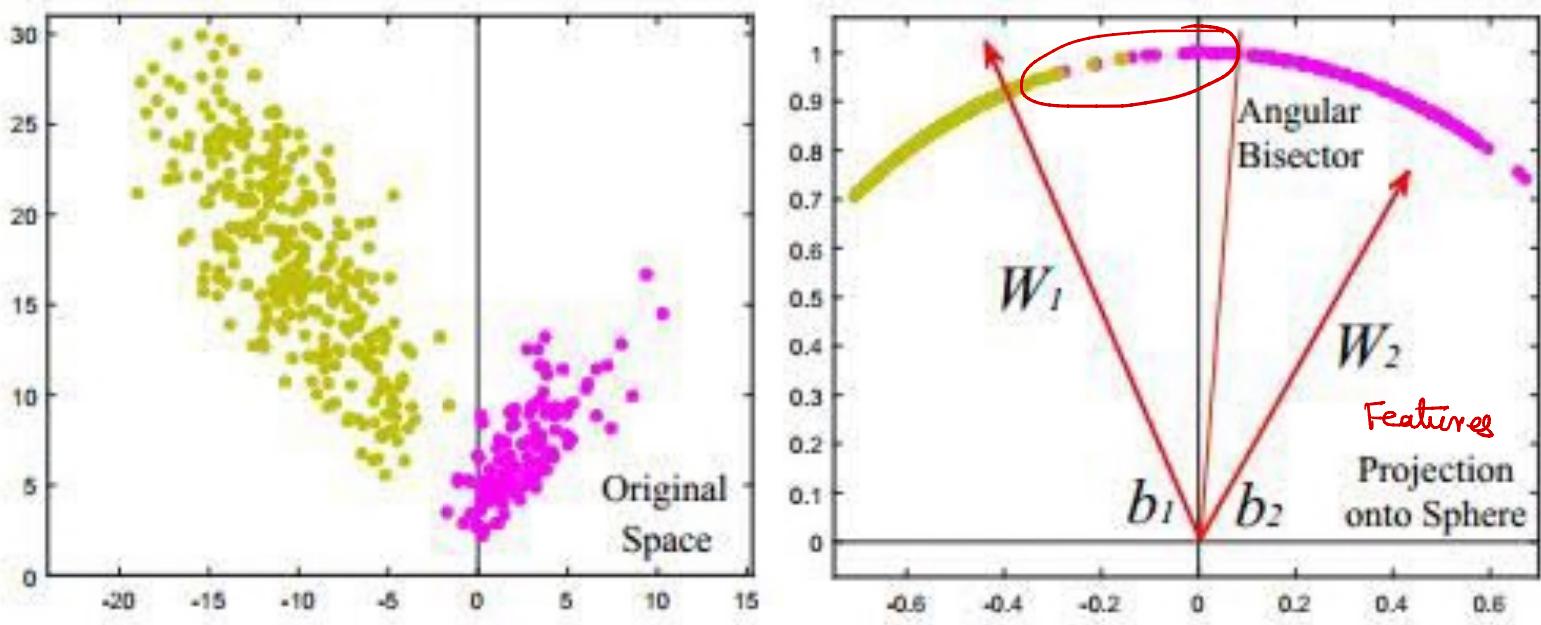
Since $(\cos \theta)$ is in $b/w (-1, 1)$ and weights are unconstrained boundaries are mostly driven by $\|W_i\| \Rightarrow$ (Bahutali)

a) $\|W_i\|$
b) $\cos(\theta_i)$ that is the angle between $[W_i]$ and $[x]$ vector.

Sphereface : Deep Hypersphere Embedding for Face Recognition.



* for classification decision boundary is learned separating the classes. This can be a serious issue when output lie near the decision boundary.



So Normal Softmax will give us infinite amount of space within some angular range (So as one can fit huge data in between)

No class compactness and inefficient feature space utilization

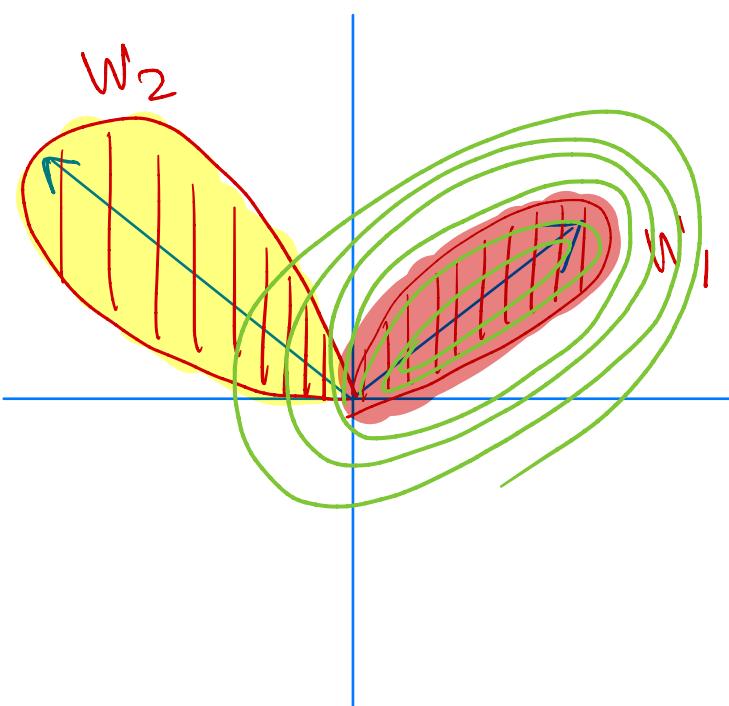
W_i 's effect

Besides, uneven and biased space partitioning

Good to capture intra class variation

But

All space got exhausted as features are spread all over



* Points "close" to (w_1) will go to class-01.

But how "close"



Need to decide by
 $\|w_1\|$ & $\|w_2\|$

This is the real nature of Softmax.

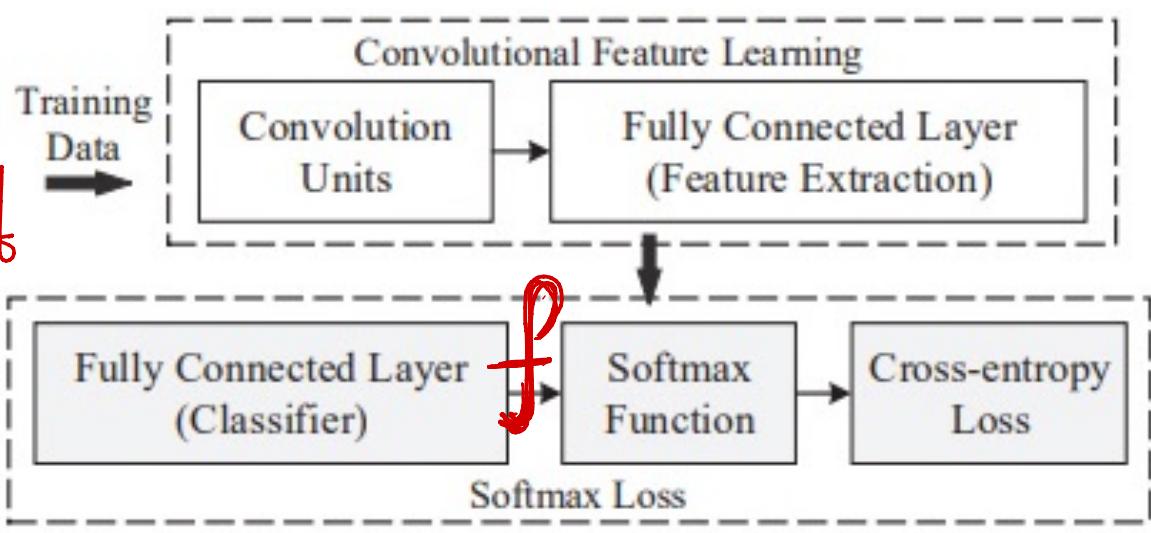
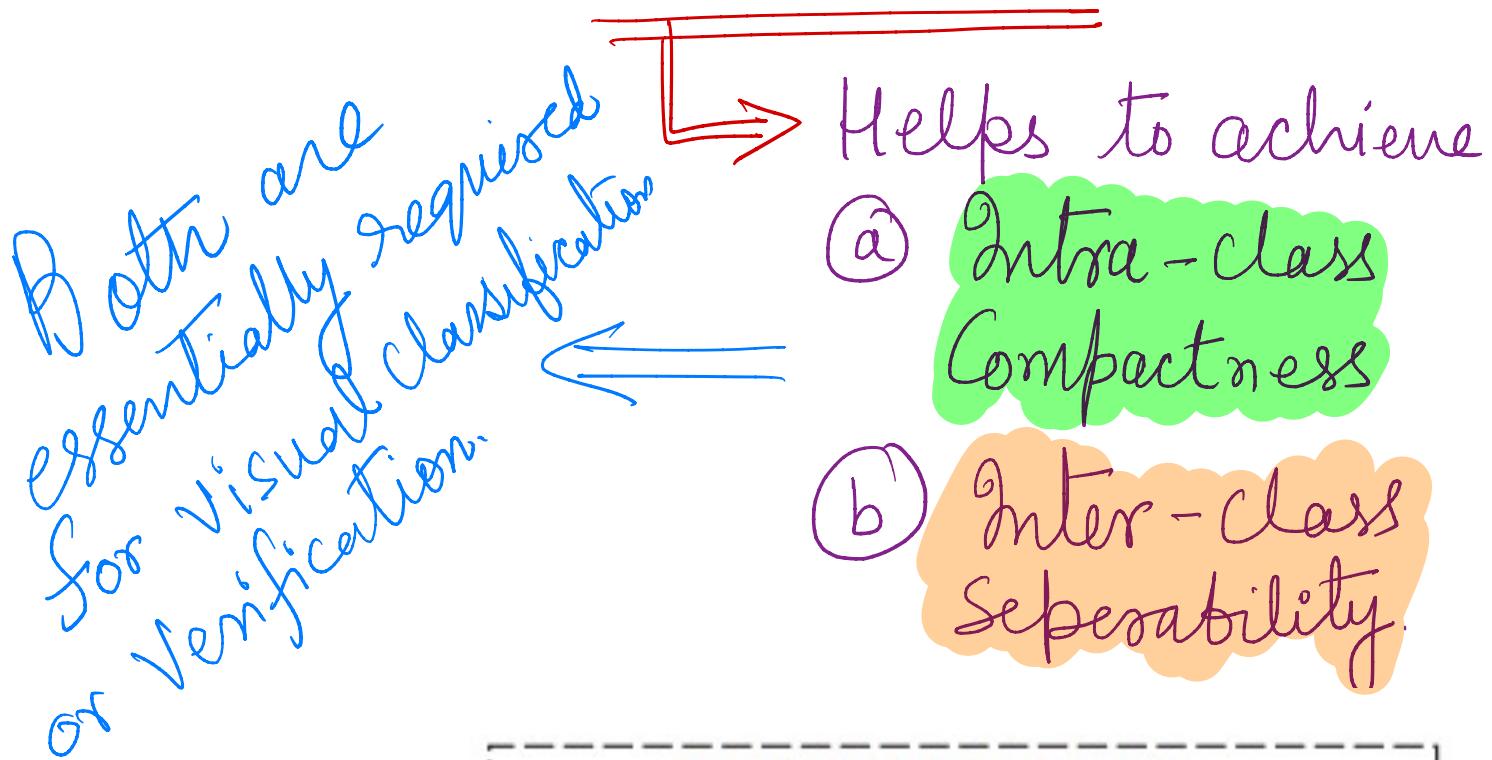
Classes are getting (within angular)
 regions
 w/o any apparent clustering &
 the N/w is free to project features
 anywhere within that space.

→ This may not be good for
 generalization and fails
 in Open - Set Scenarios.

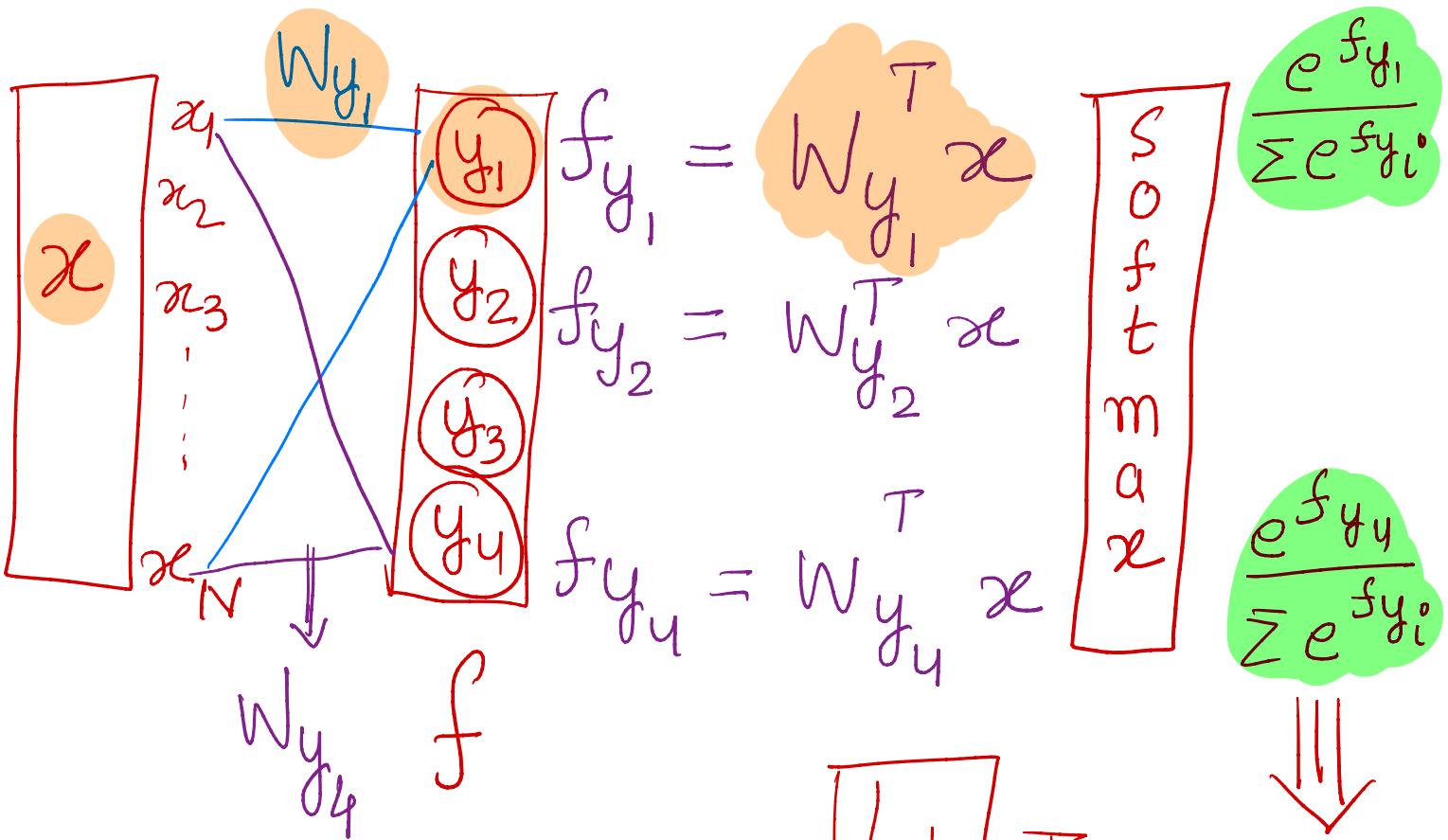
Modified Softmax

- M-S softmax to L-Softmax

Large Margin Softmax is the first paper that talks about margin similar to Triplet loss.



$f \Rightarrow$ O/P of
Last fully
connected
layer.



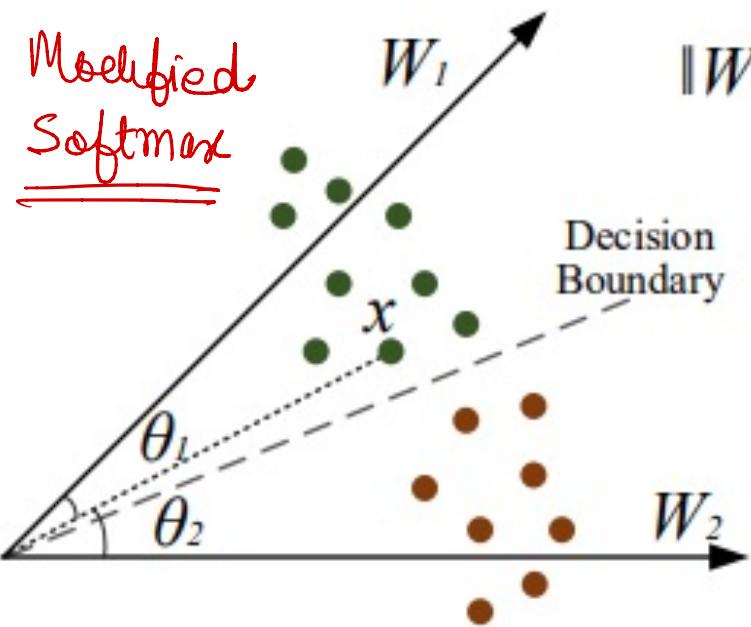
$L = -\log \frac{e^{f_{y_i}}}{\sum_j e^{f_j}}$

L_1, L_2, L_3, L_4
 They will go to
 Cross entropy
 $L = L_1 + L_2 + L_3 + L_4$

$L_i = -\log \left(\frac{e^{\|W_{y_i}\| \|x_i\| \cos(\theta_{y_i})}}{\sum_j e^{\|W_j\| \|x_i\| \cos(\theta_j)}} \right)$

Normal Softmax
 defined in terms of
 last layer O/P.

Modified Softmax



Just

Considering
Binary Classification

$$(W_1 \cdot x > W_2 \cdot x)$$

for classification of
(x) to Class - 01.

$$\|W_1\| \|x\| \cos(\theta_1) >$$

$$\|W_2\| \|x\| \cos(\theta_2)$$

Projecting weights onto unit sphere

$$\frac{W_i}{\|W_i\|}$$

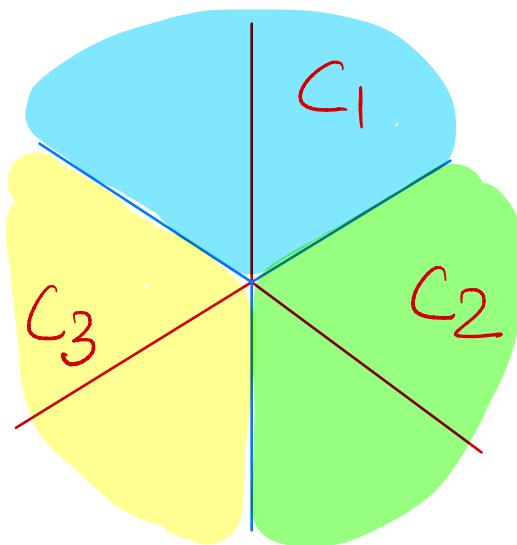
Eliminating the effect of $\|W_i\|$.
(M-Softmax)

Classification Boundary becomes

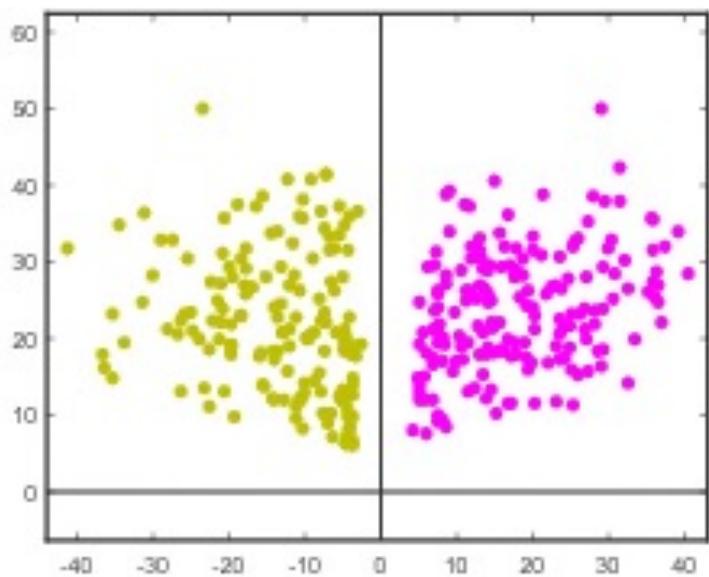
$$\Rightarrow \|x\| [\cos(\theta_1) - \cos(\theta_2)] > 0$$

This will lead to angle bisector.

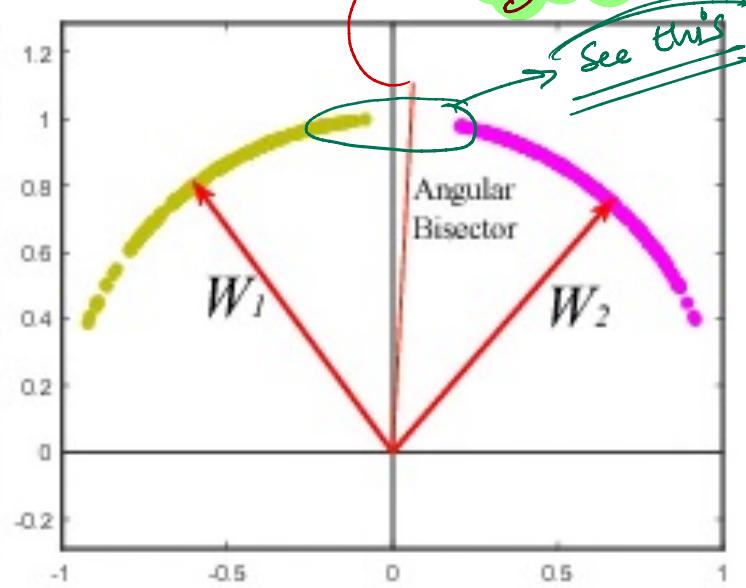
• We will have



- (a) Original feature Space.
- (b) Projected feature Space.



(c) Modified Softmax Loss

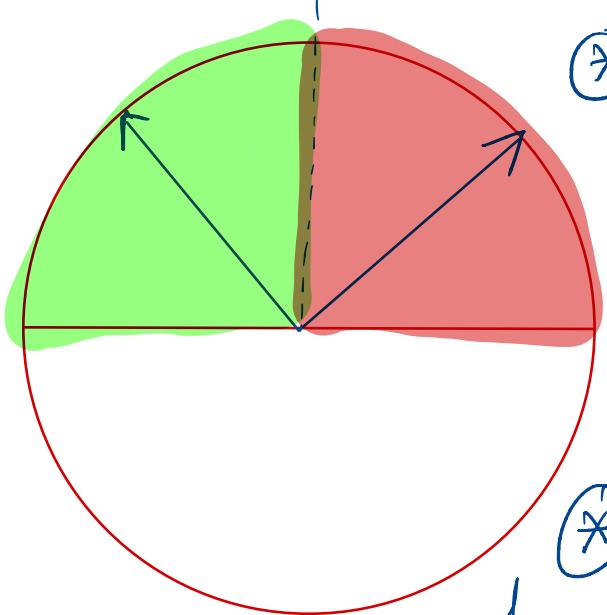


(d) Modified Softmax Loss

2-D (weights), 2-D last layer features.
all things are normalized (all or circle)

- (*) This ensures unbiased Space partitioning
- (*) Still not too much compact but better than basic Softmax

↳ Large margin
Softmax

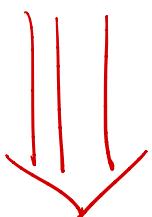


* Points are over unit Circle and angular Boundary Can be properly Seen.

* The angle is ranging from $(0 \text{ to } \pi)$

* $\cos(\theta)$ is monotonically decreasing b/w $(0 - \pi)$

i.e $[\theta \uparrow]$ then $[\cos\theta \downarrow]$



Class belonging got decreased.

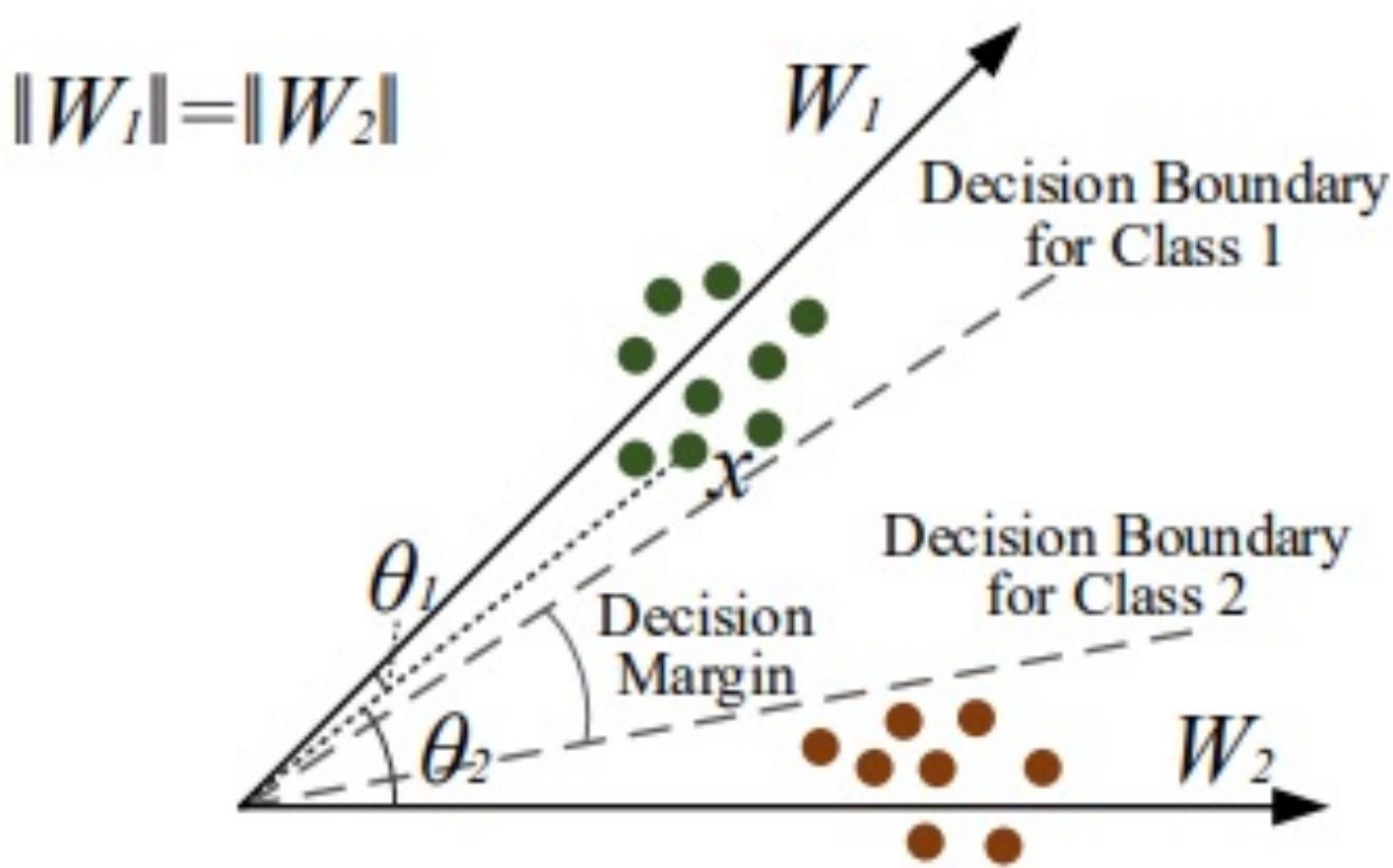
Remember that $\cos(\theta)$ is

a) monotonically decreasing $(0 \rightarrow \pi)$

b) monotonically increasing $(-\pi \rightarrow 0)$

In order to have better Class Compactness \Rightarrow Angular Margin

Angular boundaries to Angular Margins



- ④ Classification need to be Stricter with wider decision margin for better generalization.

L-Softmax

(large margin)

$$\theta$$

$$m\theta$$

angle range
($0 - \pi$)

($m \Rightarrow$ +ve integer) going to $(0 - \frac{\pi}{m})$

Now the decision boundary for class-01

requires :

$$\|\mathbf{w}_1\| \|x\| \cos(m\theta_1) >$$

$$\|\mathbf{w}_2\| \|x\| \cos(\theta_2)$$

$$L_i = -\log \left(\frac{e^{\|\mathbf{w}_{y_i}\| \|x_i\| \psi(\theta_{y_i})} \cos(m\theta_i)}{e^{\|\mathbf{w}_{y_i}\| \|x_i\| \psi(\theta_{y_i})} + \sum_{j \neq y_i} e^{\|\mathbf{w}_j\| \|x_i\| \cos(\theta_j)}} \right)$$

$$\psi(\theta) = \begin{cases} \cos(m\theta), & 0 \leq \theta \leq \frac{\pi}{m} \\ D(\theta), & \frac{\pi}{m} < \theta \leq \pi \end{cases}$$

$\textcircled{M} \rightarrow$ makes
the classification
margin larger.

$D(\theta)$ is monotonically decreasing fn

with $D(\pi/m) = \cos(\pi/m)$

$$\psi(\theta) = (-1)^k \cos(m\theta) - 2k, \quad \theta \in \left[\frac{k\pi}{m}, \frac{(k+1)\pi}{m}\right]$$

So finally $\Psi(\theta)$ is set as defined above where $k \in [0, m-1]$ k is an integer.

$$k=0$$

$$[0, \pi/m]$$

$$k=1$$

$$[\pi/m, 2\pi/m]$$

$$k=2$$

$$[2\pi/m, 3\pi/m]$$

$$\dots k=m-1$$

$$[(m-1)\pi/m, m\pi]$$

$$\Psi(\theta) =$$

$$\cos(m\theta)$$

$$-\cos(m\theta)$$

$$-2$$

$$\cos(m\theta)$$

$$-4$$

$$(-1)^{m-1} \cos(m\theta)$$

$$-2(m-1)$$

Range

$$(1, -1)$$

$$(-1, -3)$$

$$(-3, -5)$$

Hence $\Psi(\theta)$ is a monotonically decreasing continuous function depending upon $\cos(m\theta)$

Loss Function	Decision Boundary
Softmax Loss	$(\mathbf{W}_1 - \mathbf{W}_2)\mathbf{x} + b_1 - b_2 = 0$
Modified Softmax Loss	$\ \mathbf{x}\ (\cos \theta_1 - \cos \theta_2) = 0$
A-Softmax Loss	$\ \mathbf{x}\ (\cos m\theta_1 - \cos \theta_2) = 0$ for class 1 $\ \mathbf{x}\ (\cos \theta_1 - \cos m\theta_2) = 0$ for class 2

Table 1: Comparison of decision boundaries in binary case. Note that, θ_i is the angle between \mathbf{W}_i and \mathbf{x} .

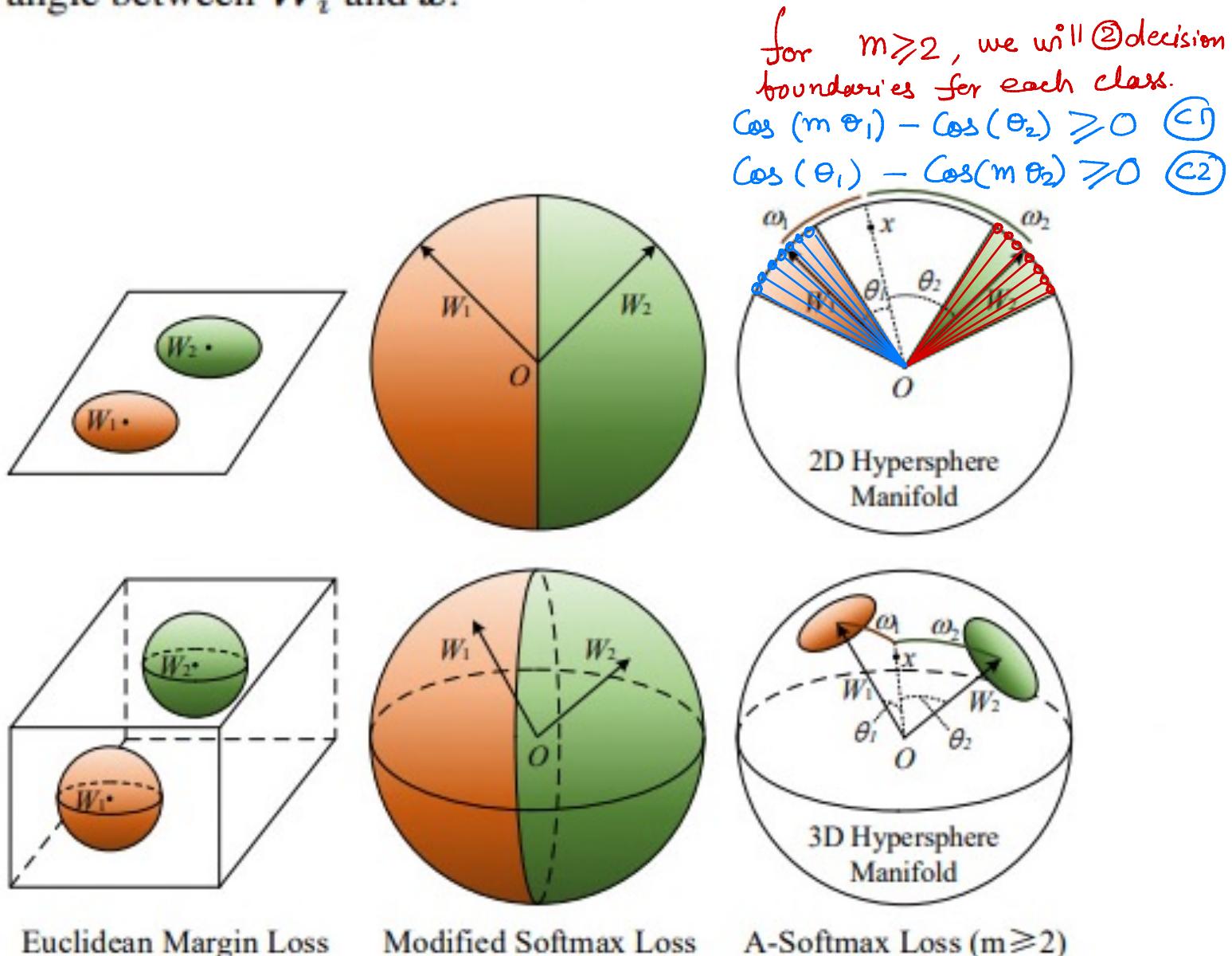


Figure 3: Geometry Interpretation of Euclidean margin loss (e.g. contrastive loss, triplet loss, center loss, etc.), modified softmax loss and A-Softmax loss. The first row is 2D feature constraint, and the second row is 3D feature constraint. The orange region indicates the discriminative constraint for class 1, while the green region is for class 2.

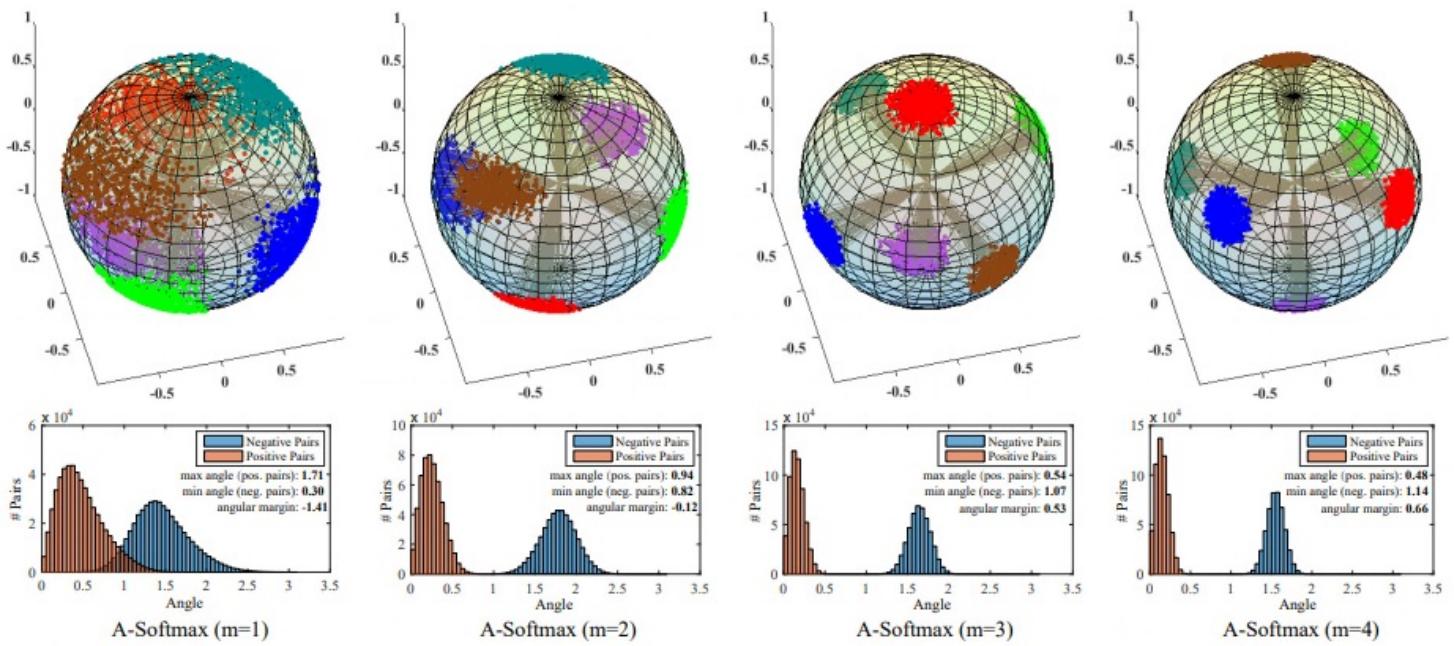
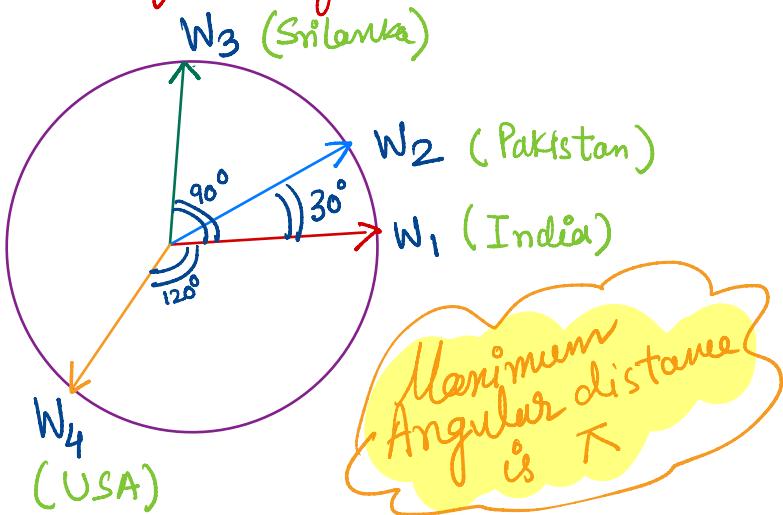


Figure 5: Visualization of features learned with different m . The first row shows the 3D features projected on the unit sphere. The projected points are the intersection points of the feature vectors and the unit sphere. The second row shows the angle distribution of both positive pairs and negative pairs (we choose class 1 and class 2 from the subset to construct positive and negative pairs). Orange area indicates positive pairs while blue indicates negative pairs. All angles are represented in radian. Note that, this visualization experiment uses a 6-class subset of the CASIA-WebFace dataset.

* In Conclusion
 Introduce the angular margin controlled by m .
 Normalize classification weights
 Zero the bias
 Lead to learn discriminative and generalized features for open set scenarios

More understanding for A-Softmax.

- ① angle of (θ_i) with the actual ground truth class (say \vec{W}_1) it will actually be considered as ($m\theta_i$) and its class likelihood further got decreased from $\cos(\theta_i) \longrightarrow \cos(m\theta_i)$



In A-Softmax Case for decision boundary : $m\theta_1 = \theta_2$

$$\text{(Binary)} \quad \& \quad \theta_1 + \theta_2 = K$$

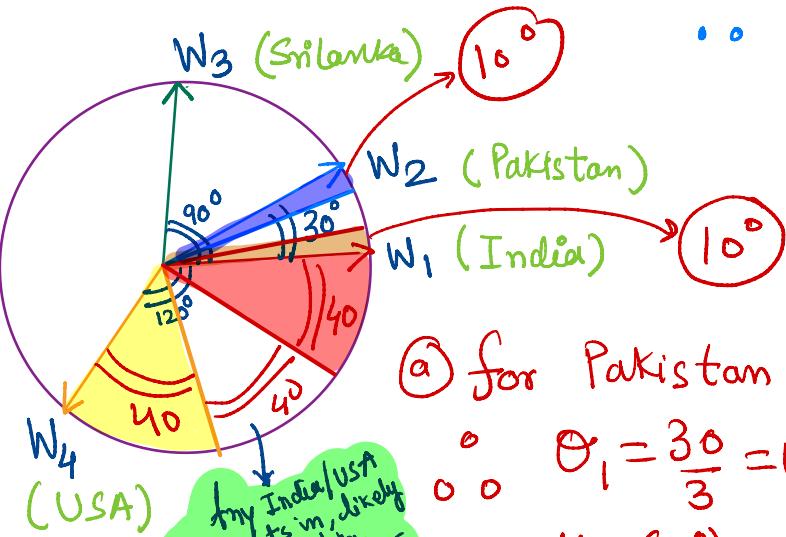
$K \Rightarrow$ angular class separation



$$\theta_1 + m\theta_1 = K$$

$$\Rightarrow [\theta_1 = K/m+1]$$

This means that for $m=2$ ($\theta_1 = K/3$) [Angular separation got divided into 3 parts]



a) for Pakistan ($K=30$)

$$\therefore \theta_1 = \frac{30}{3} = 10^\circ \text{ (for decision boundary)}$$

→ after (10°) you are not allowed to go close to Pakistan.

→ If some point with 60° = India have

$$\theta_1 = 11 \Rightarrow (\theta_2 = 19) \Rightarrow \cos(19) \checkmark$$

↓ Considered as $(22) \Rightarrow \cos(22)$

$$15 \& 15 \Rightarrow \cos(15) \checkmark$$

$$30 \Rightarrow \cos(30)$$

b) Remember you are allowed to go other side (USA), and that too upto 40° as for USA ($K=120$)

$$\therefore \theta_1 = \frac{120}{3} = 40^\circ \text{ (for dB)}$$

$$30 \rightarrow 90 \Rightarrow \cos(90) = 0$$

$$\hookrightarrow 60 \Rightarrow \cos(60)$$

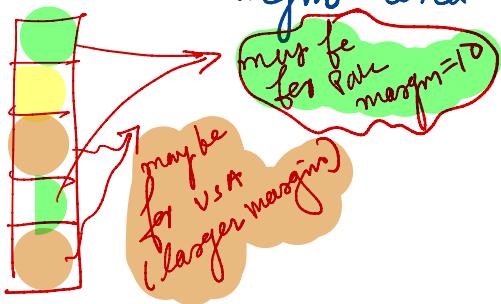
Parameter m ensures that margin between classes get maintained.

Problems with A-Softmax (Sphereface Loss)

↳ Due to integer value of (m) , the curve of target logit for GT class is very steep.
(This will hinder convergence)
(Softmax dominates)

↳ Different margins for different classes.

So in decision space some inter class features may have larger margin and others will have smaller margin.



Reduces its discrimination

CosFace :

* Decision margin is defined in Cosine space unlike Sphereface & Softmax where it is defined in angular (θ)-space.

for margin parameter (m)
assuming angular Class margin b/w

② Classes = K then

$$\text{Margin} = \left(\frac{m-1}{m+1} \right) K$$

$[m \geq 0, \text{ magnitude}]$
of Cosine margin

$$\|x\| = S$$

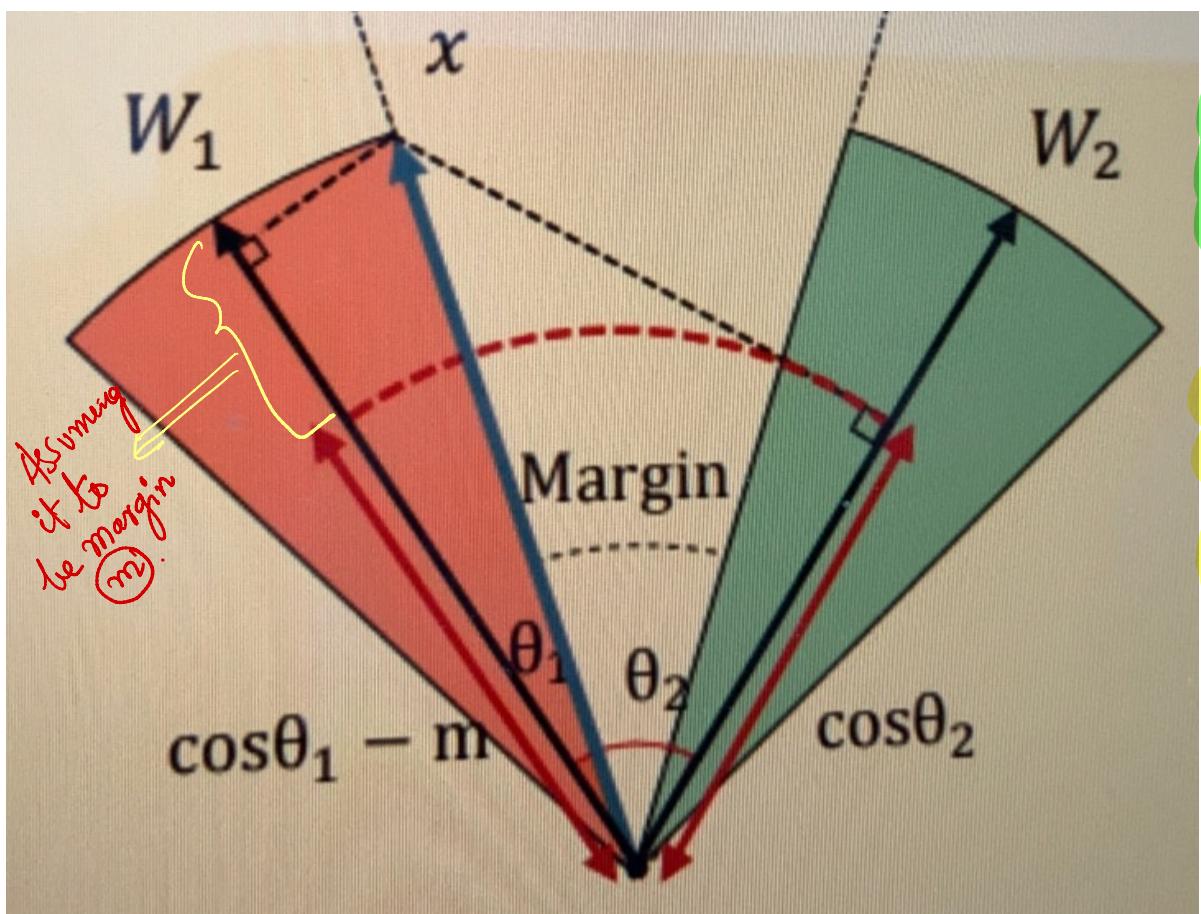
* Large Margin Cosine loss [L_{MC}]

$$L_{\text{mc}} = \frac{1}{N} \sum_i -\log \frac{e^{S(\cos(\theta_{y_i, i}) - m)}}{e^{S(\cos(\theta_{y_i, i}) - m)} + \sum_{j \neq y_i} e^{S \cos(\theta_{j, i})}}$$

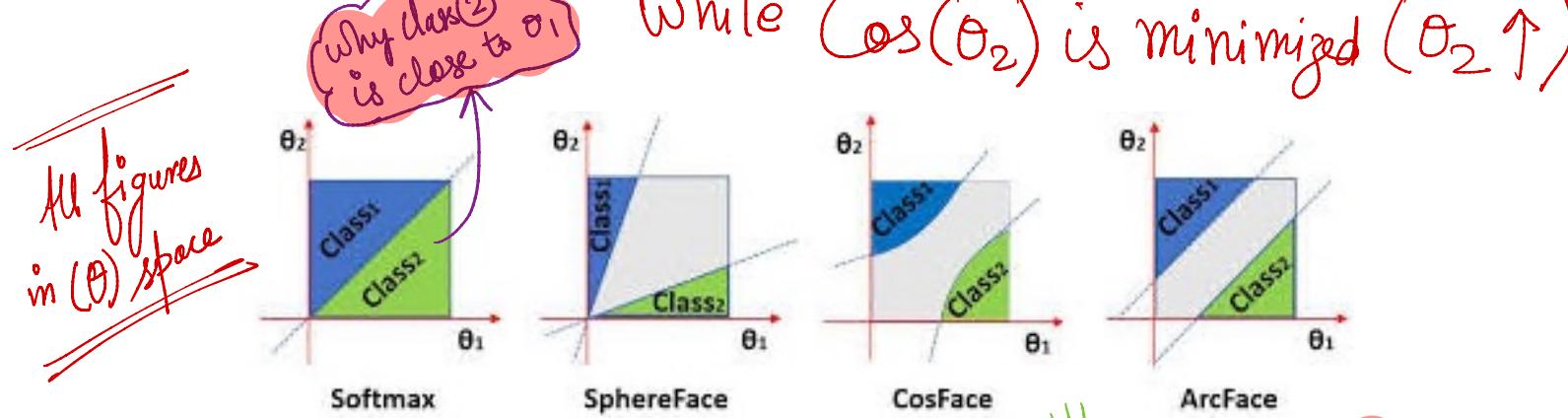
$$W = W^*/\|W^*\|$$

$$x = x^*/\|x^*\|$$

$$\cos(\theta_{j, i}) = W_j^T x_i$$



∴ for Class 1: $\cos(\theta_1)$ is maximized ($\theta_1 \downarrow$)
 while $\cos(\theta_2)$ is minimized ($\theta_2 \uparrow$)



a) $\|W_1\| \approx \|\cos \theta_1 \geq \|\|W_2\| \|\approx \|\cos \theta_2$

c) $\cos(m\theta) \geq \cos(\theta_2)$

d) $\cos(\theta_2) \geq \cos(\theta_1)$
 $\cos(\theta_1) \geq \cos(\theta_2) + m$

e) $\cos(\theta_1 + m) \geq \cos(\theta_2)$

b) $\cos \theta_1 \geq \cos \theta_2$
 Angular bisector

[At only have non-linear angular margin]

