

# RegularFace :

## Deep FaceRecognition via Exclusive Regularization

Kai Zaho, Jingyi Xu, Ming-Ming Cheng

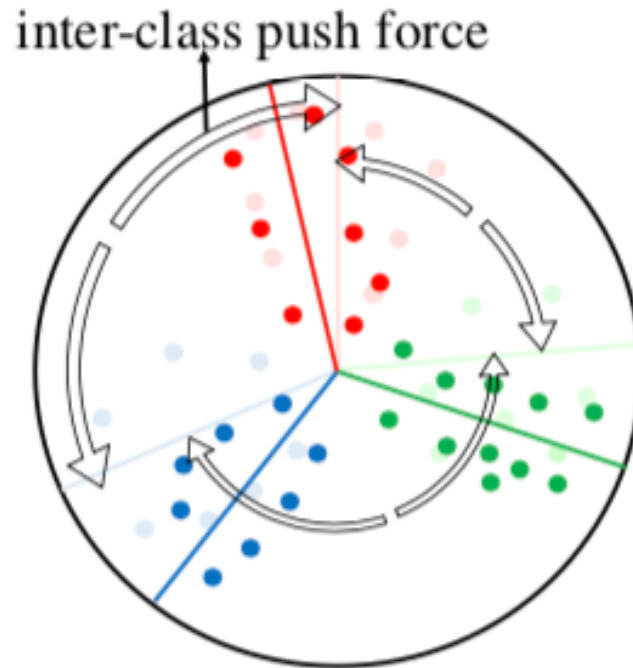
TKLNDST, CS, Nankai University

**CVPR 2019**

**Sungman, Cho.**

# Abstract

- Propose the '*exclusive regularization*' that focuses on the other aspect of discriminability : the **inter-class separability, which is neglected in many recent approaches.**
- **RegularFace explicitly distances identities** by penalizing the angle between an identity and its nearest neighbor, resulting in discriminative face representations.



(e) RegularFace

# Introduction

- Many recent works insist on **designing novel loss functions to improve the intra-class compactness** of deep features.
- Angular margin based methods focus on the **intra class compactness by clamping representations of the same identity**, either in the Euclidean space or in the sphere space.
- Exclusive regularization **explicitly enlarges the angle between parametric vectors of different identity classes**, leading to 'exclusive' classification vectors.

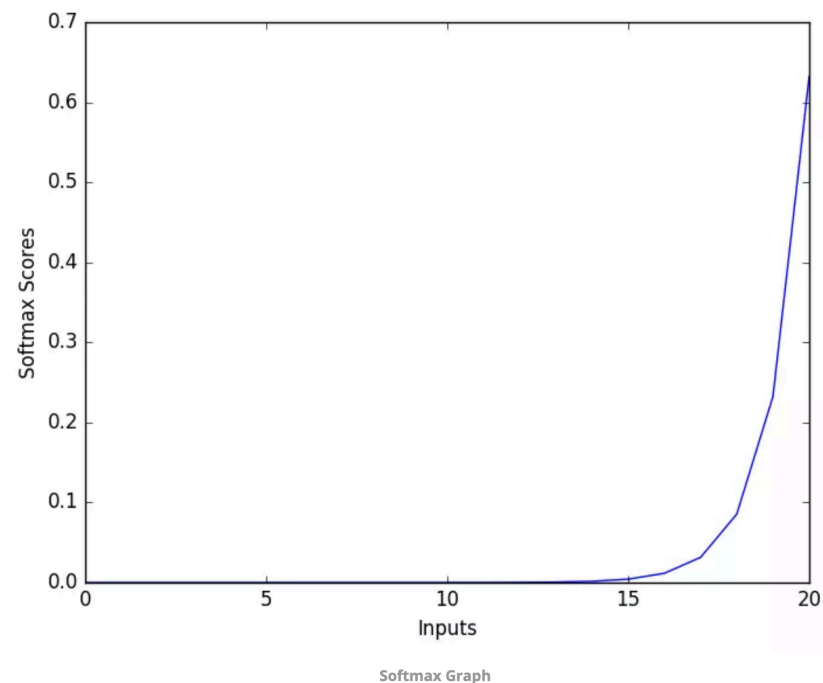
# Revisiting the Softmax

# Revisiting the Softmax (binary)

$$p_1 = \frac{\exp(\mathbf{W}_1^T \mathbf{x} + b_1)}{\exp(\mathbf{W}_1^T \mathbf{x} + b_1) + \exp(\mathbf{W}_2^T \mathbf{x} + b_2)}$$

$$p_2 = \frac{\exp(\mathbf{W}_2^T \mathbf{x} + b_2)}{\exp(\mathbf{W}_1^T \mathbf{x} + b_1) + \exp(\mathbf{W}_2^T \mathbf{x} + b_2)}$$

class 1 if  $p_1 > p_2$  and class 2 if  $p_1 < p_2$ .



# Revisiting the Softmax (binary)

$W_1^T x + b_1$  and  $W_2^T x + b_2$  determine the classification result.

Decision boundary :  $(W_1 - W_2)x + b_1 - b_2 = 0$

# Revisiting the Softmax (binary)

$W_1^T x + b_1$  and  $W_2^T x + b_2$  determine the classification result.

Decision boundary :  $(W_1 - W_2)x + b_1 - b_2 = 0$

$W_i^T x + b_i = \|W_i^T\| \|x\| \cos(\theta_i) + b_i$  ,     where  $\theta_i$  is the angle between  $W_i$  and  $x$

# Revisiting the Softmax (binary)

$W_1^T x + b_1$  and  $W_2^T x + b_2$  determine the classification result.

Decision boundary :  $(W_1 - W_2)x + b_1 - b_2 = 0$

$W_i^T x + b_i = \|W_i^T\| \|x\| \cos(\theta_i) + b_i$ , where  $\theta_i$  is the angle between  $W_i$  and  $x$



*normalization . ( $\|W_i\| = 1, b_i = 0$ )*



# Revisiting the Softmax (binary)

$W_1^T x + b_1$  and  $W_2^T x + b_2$  determine the classification result.

Decision boundary :  $(W_1 - W_2)x + b_1 - b_2 = 0$

$W_i^T x + b_i = \|W_i^T\| \|x\| \cos(\theta_i) + b_i$ , where  $\theta_i$  is the angle between  $W_i$  and  $x$



*normalization . ( $\|W_i\| = 1, b_i = 0$ )*

***posterior probabilities*** :  $p_1 = \|x\| \cos(\theta_1)$  and  $p_2 = \|x\| \cos(\theta_2)$

# Revisiting the Softmax (binary)

$W_1^T x + b_1$  and  $W_2^T x + b_2$  determine the classification result.

Decision boundary :  $(W_1 - W_2)x + b_1 - b_2 = 0$

$W_i^T x + b_i = \|W_i^T\| \|x\| \cos(\theta_i) + b_i$ , where  $\theta_i$  is the angle between  $W_i$  and  $x$



normalization . ( $\|W_i\| = 1, b_i = 0$ )

**posterior probabilities** :  $p_1 = \|x\| \cos(\theta_1)$  and  $p_2 = \|x\| \cos(\theta_2)$

same  $x$

# Revisiting the Softmax (binary)

$W_1^T x + b_1$  and  $W_2^T x + b_2$  determine the classification result.

Decision boundary :  $(W_1 - W_2)x + b_1 - b_2 = 0$

$W_i^T x + b_i = \|W_i^T\| \|x\| \cos(\theta_i) + b_i$ , where  $\theta_i$  is the angle between  $W_i$  and  $x$



normalization . ( $\|W_i\| = 1, b_i = 0$ )

**posterior probabilities** :  $p_1 = \|x\| \cos(\theta_1)$  and  $p_2 = \|x\| \cos(\theta_2)$

same  $x$

*Final result only depends on the angles  $\theta_1$  and  $\theta_2$*

# Revisiting the Softmax (binary)

$W_1^T x + b_1$  and  $W_2^T x + b_2$  determine the classification result.

Decision boundary :  $(W_1 - W_2)x + b_1 - b_2 = 0$

$W_i^T x + b_i = \|W_i^T\| \|x\| \cos(\theta_i) + b_i$ , where  $\theta_i$  is the angle between  $W_i$  and  $x$



normalization . ( $\|W_i\| = 1, b_i = 0$ )

**posterior probabilities** :  $p_1 = \|x\| \cos(\theta_1)$  and  $p_2 = \|x\| \cos(\theta_2)$

same  $x$

*Final result only depends on the angles  $\theta_1$  and  $\theta_2$*

**Decision boundary** :  $\cos(\theta_1) - \cos(\theta_2) = 0$

# Revisiting the Softmax (multi)

In multi-class case,

$$L = \frac{1}{N} \sum_i L_i = \frac{1}{N} \sum_i -\log \left( \frac{e^{f_{y_i}}}{\sum_j e^{f_j}} \right)$$

$$\begin{aligned} L_i &= -\log \left( \frac{e^{\mathbf{W}_{y_i}^T \mathbf{x}_i + b_{y_i}}}{\sum_j e^{\mathbf{W}_j^T \mathbf{x}_i + b_j}} \right) \\ &= -\log \left( \frac{e^{\|\mathbf{W}_{y_i}\| \|\mathbf{x}_i\| \cos(\theta_{y_i, i}) + b_{y_i}}}{\sum_j e^{\|\mathbf{W}_j\| \|\mathbf{x}_i\| \cos(\theta_{j, i}) + b_j}} \right) \end{aligned}$$

# Revisiting the Softmax (multi)

In multi-class case,

$$\begin{aligned} L_i &= -\log \left( \frac{e^{\mathbf{W}_{y_i}^T \mathbf{x}_i + b_{y_i}}}{\sum_j e^{\mathbf{W}_j^T \mathbf{x}_i + b_j}} \right) \\ &= -\log \left( \frac{e^{\|\mathbf{W}_{y_i}\| \|\mathbf{x}_i\| \cos(\theta_{y_i, i}) + b_{y_i}}}{\sum_j e^{\|\mathbf{W}_j\| \|\mathbf{x}_i\| \cos(\theta_{j, i}) + b_j}} \right) \end{aligned}$$

$$L_{\text{modified}} = \frac{1}{N} \sum_i -\log \left( \frac{e^{\|\mathbf{x}_i\| \cos(\theta_{y_i, i})}}{\sum_j e^{\|\mathbf{x}_i\| \cos(\theta_{j, i})}} \right)$$

# A-Softmax

Formulation, Geometric interpretation

# Introducing Angular Margin

$$L_{\text{modified}} = \frac{1}{N} \sum_i -\log \left( \frac{e^{\|\mathbf{x}_i\| \cos(\theta_{y_i, i})}}{\sum_j e^{\|\mathbf{x}_i\| \cos(\theta_{j, i})}} \right)$$



# Introducing Angular Margin

$$L_{\text{modified}} = \frac{1}{N} \sum_i -\log \left( \frac{e^{\|\mathbf{x}_i\| \cos(\theta_{y_i, i})}}{\sum_j e^{\|\mathbf{x}_i\| \cos(\theta_{j, i})}} \right)$$

making the decision more stringent


$$L_{\text{ang}} = \frac{1}{N} \sum_i -\log \left( \frac{e^{\|\mathbf{x}_i\| \cos(m\theta_{y_i, i})}}{e^{\|\mathbf{x}_i\| \cos(m\theta_{y_i, i})} + \sum_{j \neq y_i} e^{\|\mathbf{x}_i\| \cos(\theta_{j, i})}} \right)$$


# Introducing Angular Margin

$$L_{\text{modified}} = \frac{1}{N} \sum_i -\log \left( \frac{e^{\|\mathbf{x}_i\| \cos(\theta_{y_i,i})}}{\sum_j e^{\|\mathbf{x}_i\| \cos(\theta_{j,i})}} \right)$$

$$L_{\text{ang}} = \frac{1}{N} \sum_i -\log \left( \frac{e^{\|\mathbf{x}_i\| \cos(m\theta_{y_i,i})}}{e^{\|\mathbf{x}_i\| \cos(m\theta_{y_i,i})} + \sum_{j \neq y_i} e^{\|\mathbf{x}_i\| \cos(\theta_{j,i})}} \right)$$

Decision boundary:

class 1:  $\cos(m\theta_1) = \cos(\theta_2)$ .   $\theta_1 < \frac{\theta_2}{m}$

class 2:  $\cos(m\theta_2) = \cos(\theta_1)$ .   $\theta_2 < \frac{\theta_1}{m}$

# Introducing Angular Margin

$$L_{\text{modified}} = \frac{1}{N} \sum_i -\log \left( \frac{e^{\|\mathbf{x}_i\| \cos(\theta_{y_i,i})}}{\sum_j e^{\|\mathbf{x}_i\| \cos(\theta_{j,i})}} \right)$$

$$L_{\text{ang}} = \frac{1}{N} \sum_i -\log \left( \frac{e^{\|\mathbf{x}_i\| \cos(m\theta_{y_i,i})}}{e^{\|\mathbf{x}_i\| \cos(m\theta_{y_i,i})} + \sum_{j \neq y_i} e^{\|\mathbf{x}_i\| \cos(\theta_{j,i})}} \right)$$

Decision boundary:

class 1:  $\cos(m\theta_1) = \cos(\theta_2)$ .



$$\theta_1 < \frac{\theta_2}{m}$$

class 2:  $\cos(m\theta_2) = \cos(\theta_1)$ .



$$\theta_2 < \frac{\theta_1}{m}$$

A-Softmax

$$\theta_1 < \theta_2$$

$$\theta_2 < \theta_1$$

Modified Softmax

# Introducing Angular Margin

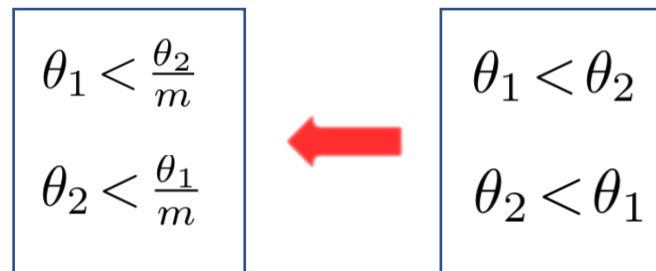
$$L_{\text{modified}} = \frac{1}{N} \sum_i -\log \left( \frac{e^{\|\mathbf{x}_i\| \cos(\theta_{y_i, i})}}{\sum_j e^{\|\mathbf{x}_i\| \cos(\theta_{j, i})}} \right)$$

$$L_{\text{ang}} = \frac{1}{N} \sum_i -\log \left( \frac{e^{\|\mathbf{x}_i\| \cos(m\theta_{y_i, i})}}{e^{\|\mathbf{x}_i\| \cos(m\theta_{y_i, i})} + \sum_{j \neq y_i} e^{\|\mathbf{x}_i\| \cos(\theta_{j, i})}} \right)$$

Decision boundary:

class 1:  $\cos(m\theta_1) = \cos(\theta_2)$ .

class 2:  $\cos(m\theta_2) = \cos(\theta_1)$ .



**more difficult  
(stringent)**

# **Inter-class Separability**

# Inter-class Separability

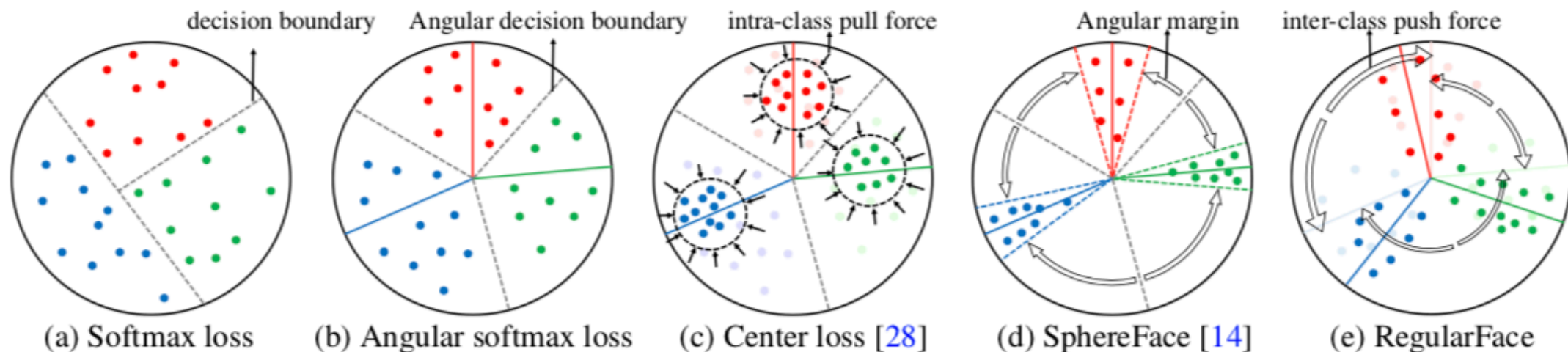
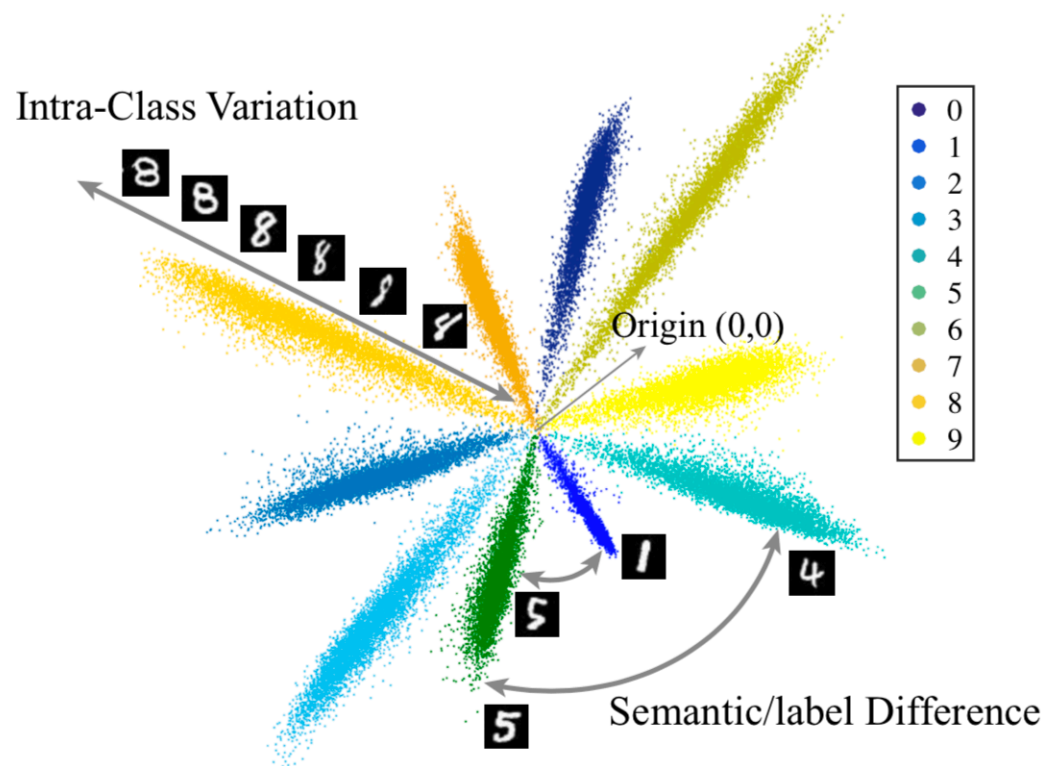


Figure 1. Illustration of face embeddings trained under various loss functions, points in color indicate different identities. (a) Softmax loss learns separable decision boundaries. (b) Angular softmax loss learns angularly separable decision boundaries. (c) *Center loss* [28] ‘pulls’ embeddings of the same identity towards their center, in order to obtain compact and discriminative representations. (d) *SphereFace* [14] (A-Softmax loss) proposes the ‘angular margin’ to clamp representations within a narrow angle. (e) Our proposed *RegularFace* introduces ‘inter-class push force’ that explicitly ‘pushes’ representations of different identities far away.

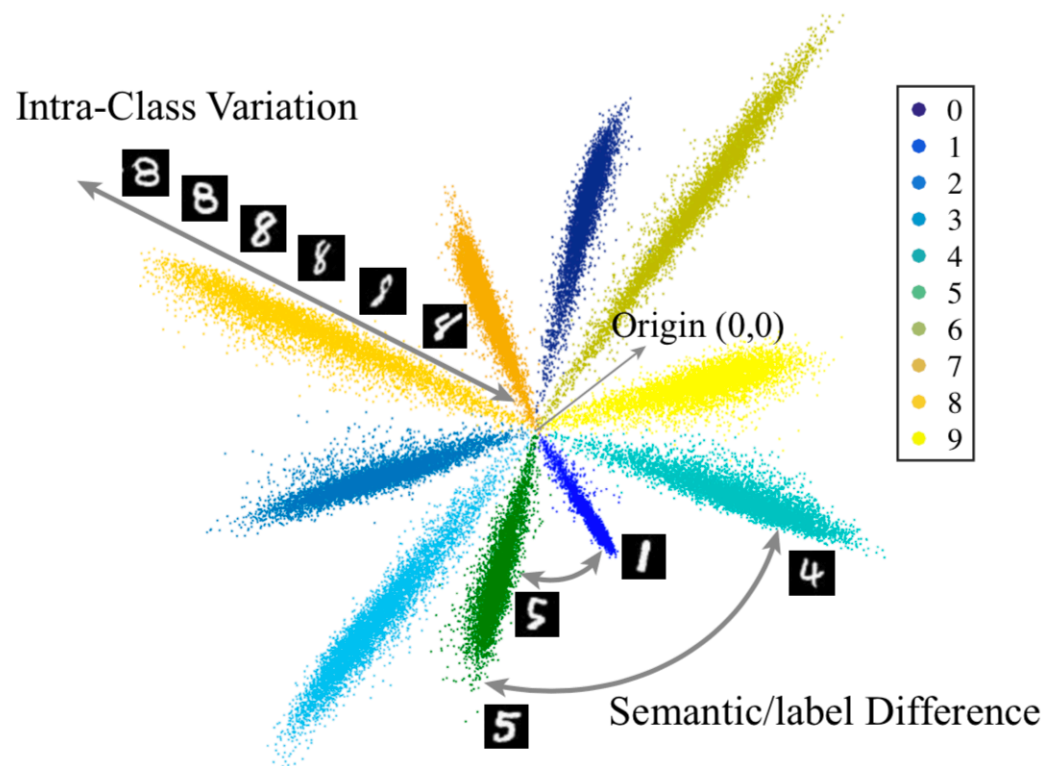
# Inter-class Separability (Motivation)

- Many recent works first perform **experiments on a tiny datasets, say MNIST**, to geometrically demonstrate the discriminability of learned representations.



# Inter-class Separability (Motivation)

- Usually, these demonstrative experiments restrict representations in a low dimension space to ease the visualization.
- In the case that **there are relatively redundant clusters than representation dimensions, the clusters tend to stretch** so as to decrease the classification error.

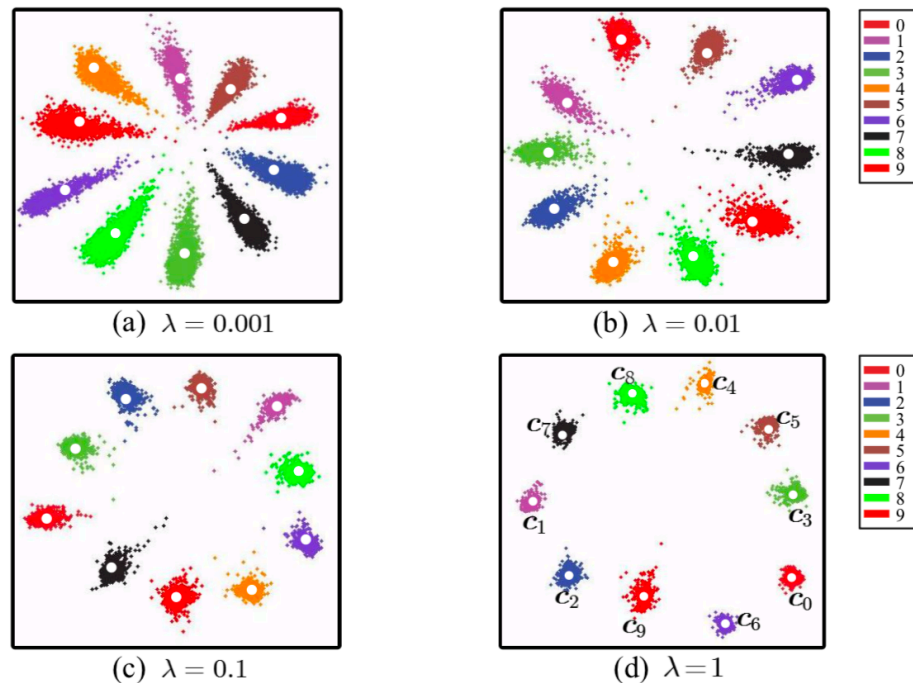




**How can we measure  
inter-class separability ?**

# Inter-class Separability (Motivation)

- The cluster centers of representations are nearly uniformly distributed and hold the maximal inter-cluster distances in a 2D plane (Fig. Center Loss)
- In face recognition task, the cluster centers may not be so well distributed. (512 dimension  $\rightarrow$  10K identities)



# Inter-class Separability (Motivation)

$$Sep_i = \max_{j \neq i} \cos(\varphi_{i,j}) \quad \text{angle between } W_i \text{ and } W_j$$
$$= \max_{j \neq i} \frac{W_i \cdot W_j}{\|W_i\| \cdot \|W_j\|},$$

- Ideally the cluster centers are expected to be **uniformly distributed** and be as far away (small cosine value) from others as possible.
- In other words  $\text{mean}(Sep)$  and  $\text{std}(Sep)$  are expected to be as small as possible.

# Inter-class Separability (Motivation)

Methods	mean( $Sep$ )	std( $Sep$ )
Softmax Loss	0.286	0.0409
Center Loss[28]	0.170	0.134
SphereFace[14]	0.170	0.013
Random	0.16992	0.027

Table 1. Inter-class separability of different models. ‘Random’ means the model parameters are draw from a uniform distribution.

**Cluster centers in existing methods are not so well distributed.**

# Inter-class Separability (Motivation)

Methods	mean( $Sep$ )	std( $Sep$ )
Softmax Loss	0.286	0.0409
Center Loss[28]	0.170	0.134
SphereFace[14]	0.170	0.013
Random	0.16992	0.027

Table 1. Inter-class separability of different models. ‘Random’ means the model parameters are draw from a uniform distribution.

Cluster centers in existing methods are not so well distributed.

→ Potentially improve feature discrimination by enhancing the inter-class separability.

# **Exclusive Regularization**

# Exclusive Regularization

Angular softmax loss

$$\mathcal{L}_s(\theta, W) = \frac{1}{N} \sum_{i=1}^N -\log \frac{e^{\|x_i\|_2 \cos(\phi_{i,y_i})}}{\sum_j e^{\|x_i\|_2 \cos(\phi_{i,j})}},$$

# Exclusive Regularization

Angular softmax loss:

$$\mathcal{L}_s(\theta, W) = \frac{1}{N} \sum_{i=1}^N -\log \frac{e^{\|x_i\|_2 \cos(\phi_{i,y_i})}}{\sum_j e^{\|x_i\|_2 \cos(\phi_{i,j})}},$$

Exclusive Regularization:

$$\mathcal{L}_r(W) = \frac{1}{C} \sum_i \max_{j \neq i} \frac{W_i \cdot W_j}{\|W_i\| \cdot \|W_j\|}.$$

Overall loss function:

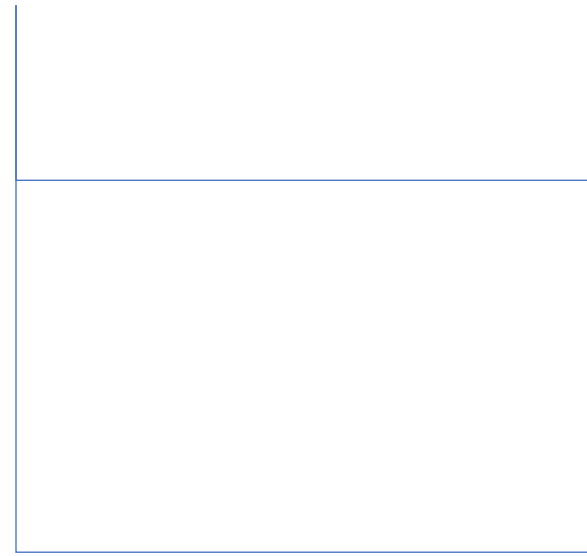
$$\mathcal{L}(\theta, W) = \mathcal{L}_s(\theta, W) + \lambda \mathcal{L}_r(W),$$

**Inter-class push force + intra-class pull force.**



# Optimize with Projected Gradient Descent

$$(\theta^*, W^*) = \underset{(\theta, W)}{\operatorname{argmin}} \mathcal{L}(\theta, W).$$


$$\theta^{t+1} = \theta^t - \alpha \frac{\partial \mathcal{L}_s(\theta^t, W)}{\partial \theta^t},$$

$$\begin{cases} \hat{W}^{(t+1)} = W^t - \alpha \frac{\partial \mathcal{L}}{\partial W^t} \\ W^{(t+1)} = \text{Normalize}(\hat{W}^{(t+1)}). \end{cases}$$

# Gradient of Exclusive Regularization

$$\frac{\partial \mathcal{L}_r(W)}{\partial W_j} = W_{j'} + \sum_{W_i \in \mathbb{C}} W_i$$

where  $W_{j'}$  is the nearest neighbor of  $W_j$ :

$$j' = \operatorname{argmax}_{i \in \{1, \dots, C\}, i \neq j} W_j \cdot W_i.$$

$\mathbb{C}$  is the collection of columns whose nearest neighbor is  $W_j$ :

$$\forall W_i \in \mathbb{C}, \quad \operatorname{argmax}_{k \in \{1, \dots, C\}, k \neq i} W_i \cdot W_k = j.$$

# Architecture

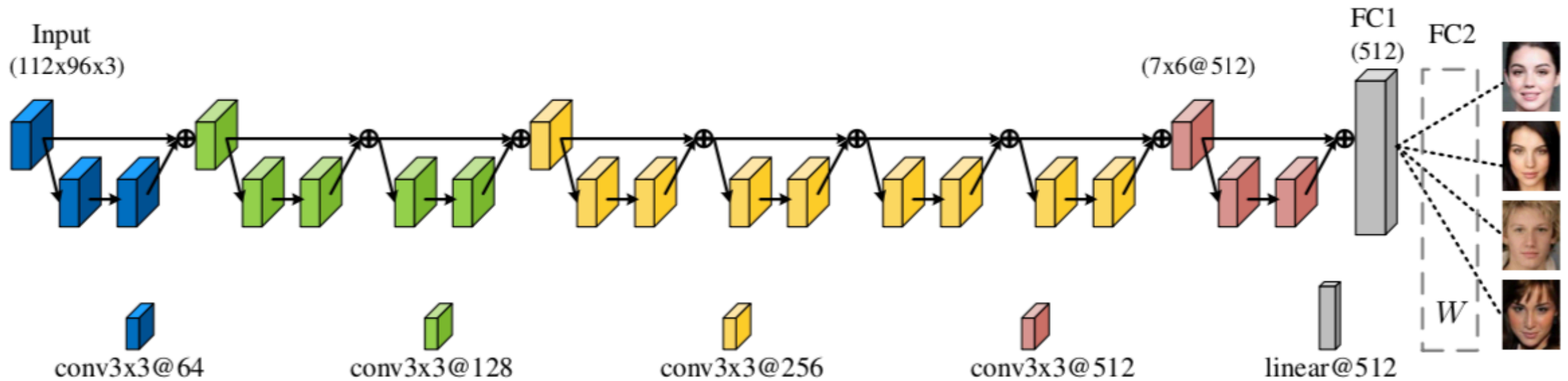
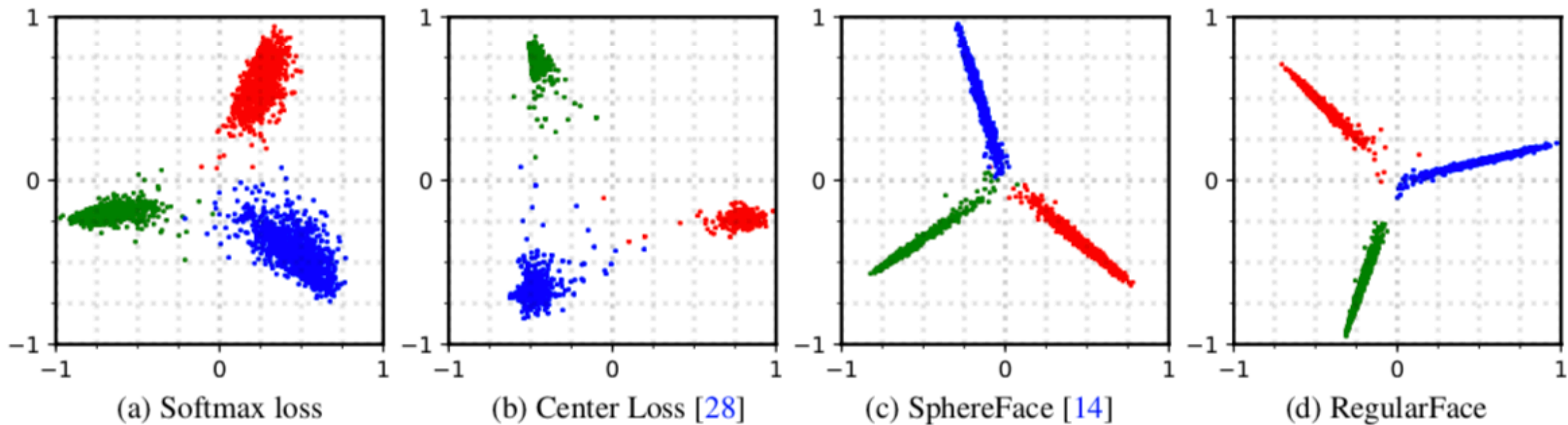


Figure 2. The ResNet20 architecture. 'conv3x3@ $\mathcal{X}$ ' represents a  $3 \times 3$  convolutional layer that outputs  $\mathcal{X}$  feature maps, and  $\oplus$  represents element-wise sum.  $W$  is a matrix that maps the facial representation to probabilities of input image belonging to identities.

# Experiments



# Experiments

Method	Data	LFW	YTF
DeepFace [26] (3)	4M	97.35	91.4
FaceNet [22]	4M	<b>99.65</b>	<b>95.1</b>
DeepID2+ [24]	4M	98.70	-
DeepID2+ [24] (25)	4M	99.47	93.2
Center Loss [28]	0.7M	99.28	94.9
Softmax Loss (SM)	WebFace	97.88	90.1
Center Loss [28]		98.91	93.4
L-Softmax [15]		99.01	93.0
SphereFace [14]		99.26	94.1
<b>RegularFace+SM</b>		99.02	91.9
<b>RegularFace+[28]</b>		99.18	93.7
<b>RegularFace+[14]</b>		<b>99.33</b>	<b>94.4</b>
Softmax Loss (SM)	VGGFace2	98.55	93.4
Center Loss [28]		99.31	94.3
L-Softmax [15]		99.35	94.1
SphereFace [14]		99.50	95.9
<b>RegularFace+SM</b>		99.32	94.7
<b>RegularFace+[28]</b>		99.39	95.1
<b>RegularFace+[14]</b>		<b>99.61</b>	<b>96.7</b>

Method	Protocol	Rank1 Acc	Ver.
Softmax loss (SM)	Small	52.86	65.93
L-Softmax [15]		67.13	80.42
Center Loss [28]		65.23	76.52
SphereFace [14]		69.62	83.16
<b>RegularFace+SM</b>		65.91	78.21
<b>RegularFace+[28]</b>		68.37	81.25
<b>RegularFace+[14]</b>		<b>70.23</b>	<b>84.07</b>
Softmax loss(SM)	Large	61.72	70.52
Center Loss [28]		70.29	87.01
SphereFace [14]		74.82	89.01
<b>RegularFace+SM</b>		72.91	88.37
<b>RegularFace+[28]</b>		73.27	89.14
<b>RegularFace+[14]</b>		<b>75.61</b>	<b>91.13</b>

**Thank You.**