

## Introduction

- 기존 연구(Center loss, SphereFace, CosFace, ArcFace 등)들은 Intra-class 의 compactness 를 맞추는데 중점이 되었다.
- 본 논문에서는 inter class 를 잘 구분하도록 네트워크를 설계하는 방법에 대하여 제안하였고, 이 방법이 기존 연구들에서는 제안되지 않았던 형태라고 언급한다.
- '*exclusive regularization*' 을 classification layer 에 적용해서, 서로 다른 클래스들 간의 간격을 크게 만들어 준다.

## Obeservation and Motivation

- RegularFace 는 Center-Loss 와 SphereFace 에서 영감을 받아 만들어졌다.
- **Center-Loss**
  - embedding 된 features 와 그 중심을 제한하여, intra-class 를 compactness 하게 만들어준다.

$$L_{center} = \frac{1}{2} \sum_{i=1}^N \|x_i - c_{y_i}\|_2^2$$

- $x_i \in R^K$  는 샘플  $i$  의 feature embedding 을 나타내고,  $c_{y_i}$  는 label이  $y_i$  인 샘플의 embedding center 를 나타낸다.
- **Softmax Loss & Angular Softmax Loss**
  - Softmax Loss

$$p_c(x_i) = \frac{e^{W_c^T x_i + b_c}}{\sum_{j=1}^C e^{W_j^T x_i + b_j}}$$

- Angular Softmax Loss

$$p_c(x_i) = \frac{e^{\|x_i\| \cos(\phi_{i,c})}}{\sum_{j=1}^C e^{\|x_i\| \cos(\phi_{i,j})}}$$

- $\phi_{i,j}$  은 feature embedding  $x_i$  와 weight vector  $W_j$  의 각도를 나타낸다. softmax-loss 를 최소화 시키는 것은  $\phi_{i,j}$  를 최소화시키는 문제와 동일하기 때문에,  $W_j$  는 모든  $x_i$  의 클러스터 중심으로 생각할 수 있다. ( $y_i = j$  일때)

- **SphereFace**

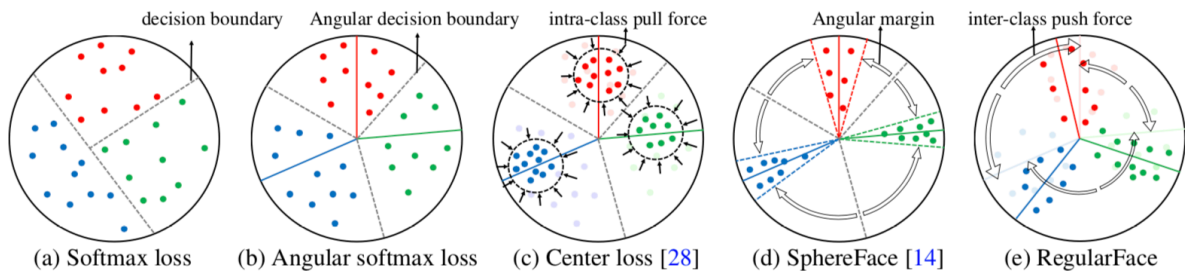
- SphereFace 는 angular softmax loss 에서 angular margin 을 추가한 개념이다. 수식은 아래와 같다.

$$p_c(x_i) = \frac{e^{\|x_i\| \cos(m \cdot \phi_{i,y_i})}}{e^{\|x_i\| \cos(m \cdot \phi_{i,y_i})} + \sum_{j \neq y_i} e^{\|x_i\| \cos(\phi_{i,j})}}$$

- $m \in \mathbb{Z}_+ = \{1, 2, \dots\}$  은 margin 을 조절하는 파라미터이다.  $m$  이 1일 경우에, Angular Softmax Loss 와 식이 같아진다.

- **Feature Embeddings**

아래 그림은 다양한 Loss function에 따른 Feature embedding을 보여준다.



## Inter-class Separability

- Inter-class 를 잘 나누는 것과, intra-class 를 잘 뭉치게 하는것은 차별성을 만드는 중요한 요소이다. 하지만, Sphreface, Center loss 등의 기존 방법은 intra-class 를 잘 뭉치게 하는 방법에 집중했다.
- 이전의 실험들은 시각화를 위해 MNIST 데이터를 사용했다. 이 경우에는 representation dimension 에 비해 redundant cluster 가 있는 것을 확인할 수 있다. cluster 들은 classification error를 줄이기 위해 퍼진 형태를 띄는 경향이 있다. (저자는 2D Visualization 에서 Center loss 를 확인해보면, cluster center 들이 uniform 하게 배치되어 있는 것을 확인할 수 있는데, 실제 얼굴인식 테스트 (512d vector로 10K identity 구별)에서는 cluster center 가 잘 배치될 수 없다고 주장한다.)
- Inter-class separability 를 측정하기 위해서 다음과 같은 수식을 제안한다.

( $\phi_{i,j}$  는  $W_i$  와  $W_j$  사이의 각도를 나타낸다.)

$$\begin{aligned} Sep_i &= \max_{j \neq i} \cos(\phi_{i,j}) \\ &= \max_{j \neq i} \frac{W_i \cdot W_j}{\|W_i\| \cdot \|W_j\|} \end{aligned}$$

- Cluster center 들은 uniformly distributed 하며, 최대한 서로 멀리 떨어져 있는 상태가 이상적이다. 최대한 서로 멀리 떨어져 있는 상태는 cos 값이 최소일 때를 나타낸다. (논문에서는 직접적으로 언급하고 있지 않지만 cos 을 0부터  $\pi$  값으로 제약시켜둔 것 같다. 대부분의 논문에서도 cos 값은 0부터  $\pi$  까지로 제약하여 단조감소 함수 형태로 만든다.)

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

- 위의 표를 보면, 기존의 연구들이 생각보다 cluster center 를 잘 분포시키지 못한다는 것을 확인할 수 있다. 이를 기반으로 저자는 cluster center 를 조금 더 잘 분포시킬 수 있는 방법이 효과가 있을 것이라고 판단하였고, 'exclusive regularization' 을 제안하였다.

## Exclusive Regularization

- Angular Softmax Loss

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

$\phi_{i,j}$  은 feature embedding  $x_i$  와 weight vector  $W_j$  의 각도를 나타낸다.

- Exclusive Regularization

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

- Overall Lossfunction

$$L(\theta, W) = L_s(\theta, W) + \lambda L_r(W)$$

- $L_s$  는 A-softmax, center loss 등 다양한 Loss function 으로 대체될 수 있다.
- 위의 두가지 loss function 을 결합함으로써, inter-class push force 와 intra-class pull force 두 가지의 효과를 얻을 수 있다.

## Optimize with Projected Gradient Descent

- Optimization

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

- For  $\theta$  update,

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

- For  $W$  update,

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

- 위의 수식에서 Normalization 부분은,  $W$  를 다시 hypersphere 상에 projection 시키는 의미를 지니며, 'project step' 이라고 부른다.  $W$  는 sphere 표면에 존재하기 때문에, L2 norm 을 사용하였다.

## Architecture

- 네트워크는 ResNet 구조를 약간 변형한 ResNet-20 을 사용하였다. FC1 은 512d vector를 뽑아내기 위한 구조이고, FC2는 Classification 을 하기 위한 구조이다.

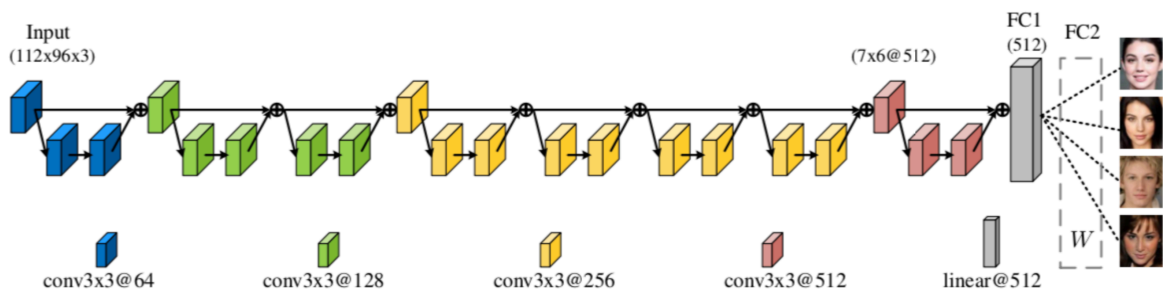
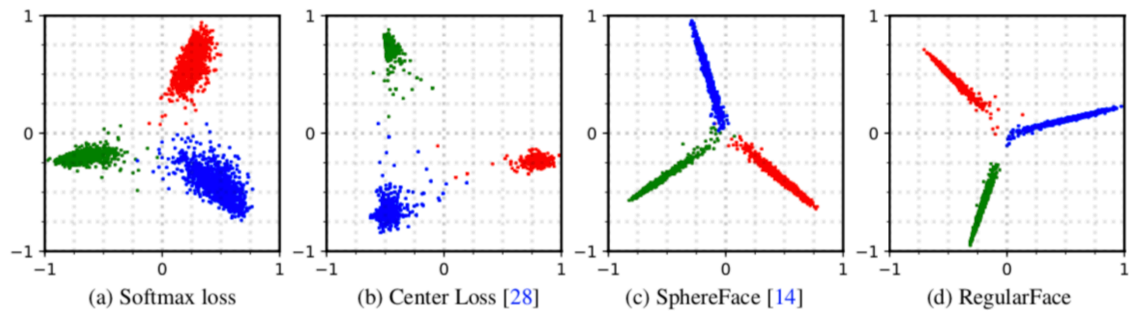


Figure 2. The ResNet20 architecture. 'conv3x3@ $\mathcal{X}$ ' represents a  $3 \times 3$  convolutional layer that outputs  $\mathcal{X}$  feature maps, and  $\oplus$  represents element-wise sum.  $W$  is a matrix that maps the facial representation to probabilities of input image belonging to identities.

## Experiments

- **Training data**
  - CASIA-WebFace, VGGFace2
- **Preprocessing**
  - MTCNN 을 활용하여 112x96 으로 crop 하였고, alignment를 위해서 모든 사진의 눈의 위치를 고정시키는 방법을 사용하였다.
- **Evaluation Protocol**
  - LFW, YTF, MegaFace challenge을 사용하였다.
  - 각 이미지에서 horizontal flip을 사용하여 1024 vector를 추출하여 사용하였다.
  - LFW, YTF 데이터셋은 이미지 간의 코사인 유사도를 계산하는 방식을 사용해서 10-fold-cross-validation으로 측정하였다.
  - MegaFace challenge 는 challenge 에서 제공하는 공식 평가 툴을 사용하여 측정하였다.
- **Different Loss Formulas**
  - 이전에 언급했던 것 처럼 anuglar softmax loss 부분은 다양한 기존 알고리즘으로 대체가 가능하다. 아래 그림은 MNIST 에서 3개의 digit을 뽑아 다양한 기존 알고리즘을 실험해본 결과이다.



- 저자는 앞서 언급했던 redundant cluster 문제를 언급하면서, MNIST 실험을 Face recognition task 와 비슷하게 하기 위하여 LeNet을 사용하고 MNIST 중 3개의 digit 만을 뽑아서 학습했다고 말한다.

## • LFW, YTF 데이터셋

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>

## • MegaFace Challenge

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>

## Conclusion

- 기존 연구들에서는 intra-class 를 compact 하게 만드는데에 집중한 반면에, RegularFace 에서는 'Exclusive regularization' 를 활용하여 explicit 하게 inter-class의 거리를 증가시켰다.