ArcFace: Additive Angular Margin Loss for Deep Face Recognition

Published in CVPR 2019

* Material collected from various sources.
No authorship is claimed.

Surbhi Mittal
P19CS208

What is ArcFace?

- Loss function specifically designed for Face Recognition

Face Recognition

Performing Face Recognition

- Classification problem but with a large number of classes.
- Thousands of identities depending on the dataset

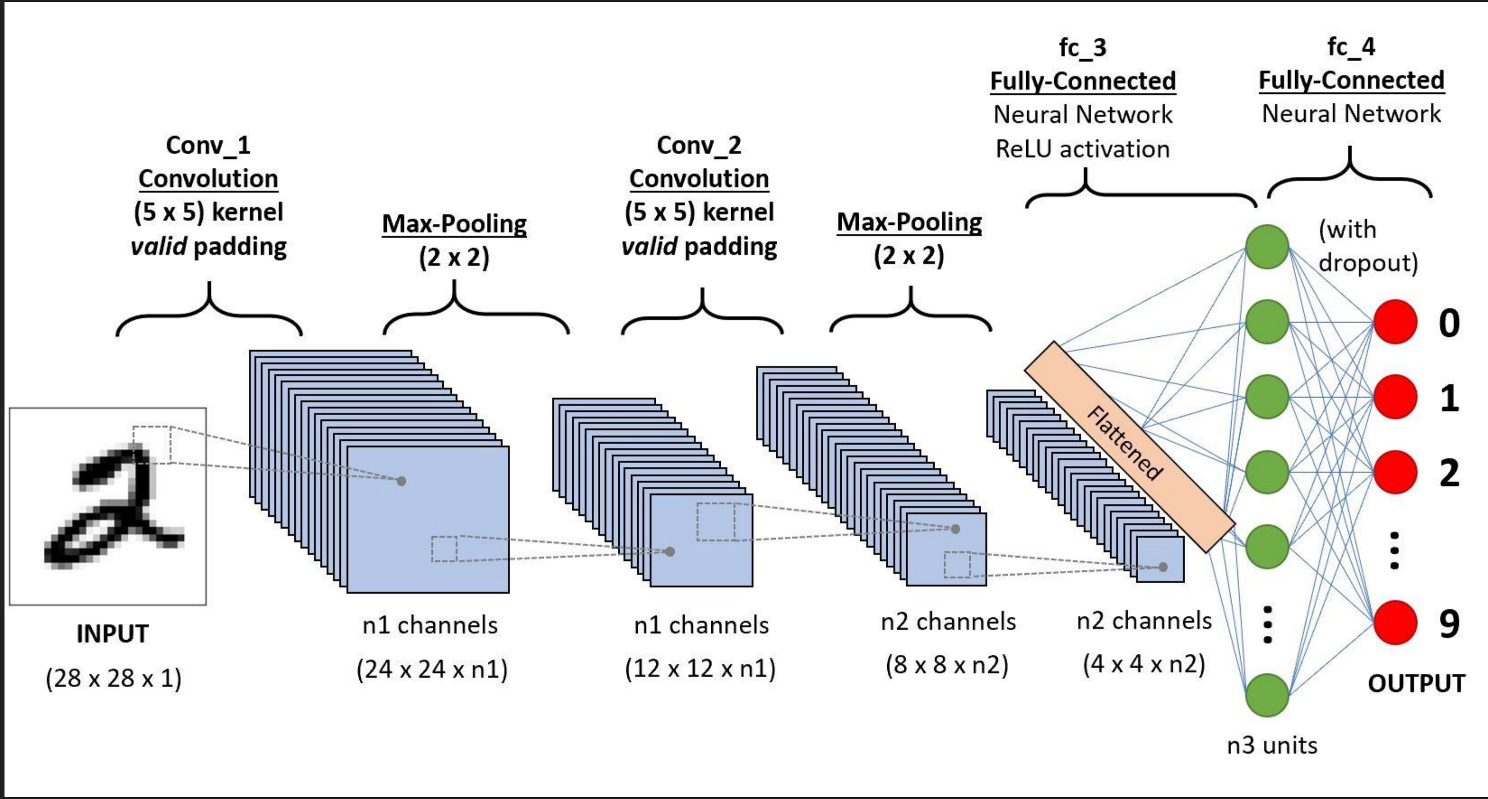
Aim-

- Learn feature embeddings

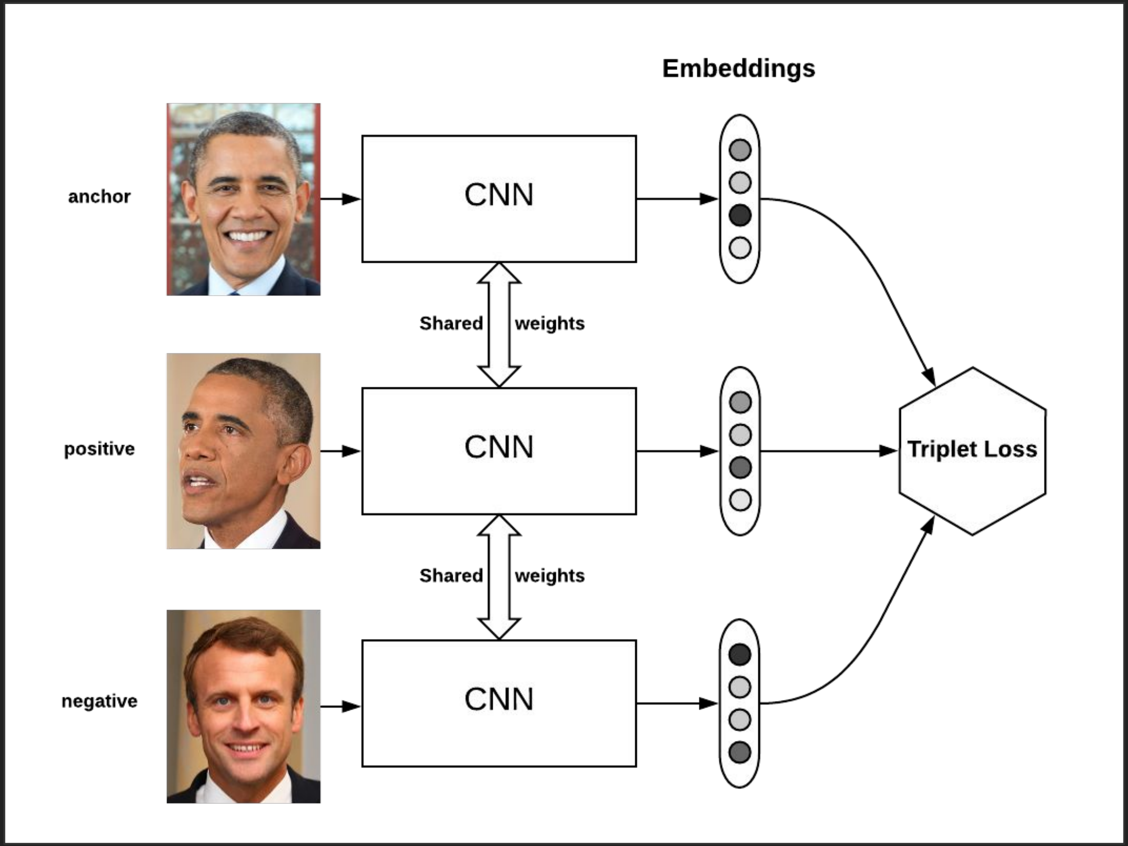
Popular Approaches-

- Softmax Loss
- Direct Feature Embeddings

Softmax Loss



Feature Embeddings using Triplet Loss



Limitations

For the softmax loss, the size of the linear transformation matrix $W \in R_{d\times n}$ increases linearly with the identities number n.

$$L_1 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}},$$

For the triplet loss, there is a combinatorial explosion in the number of face triplets especially for large-scale datasets, leading to a significant increase in the number of iteration steps.

Centre Loss

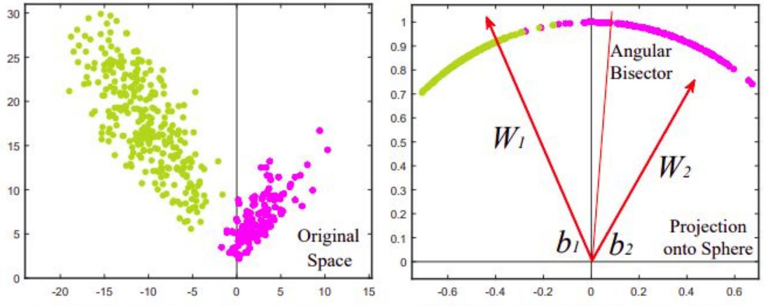
Wen et al. pioneered the centre loss, the Euclidean distance between each feature vector and its class centre, to obtain intra-class compactness while the interclass dispersion is guaranteed by the joint penalisation of the softmax loss.

$$\mathcal{L}_C = \frac{1}{2}\sum_{i=1}^{m} \|\boldsymbol{x}_i - \boldsymbol{c}_{y_i}\|_2^2$$

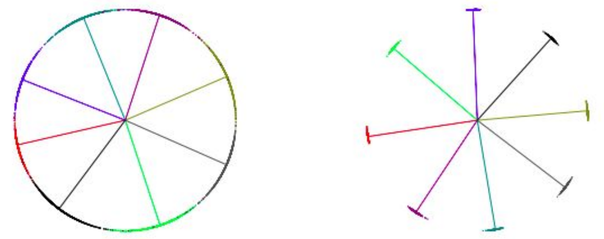
However, with large number of classes-

- Calculating centres is computationally expensive.
- Centres need to be created and redefined in each batch.
- Uses euclidean distance.

Euclidean Distance *vs* Geodesic Distance

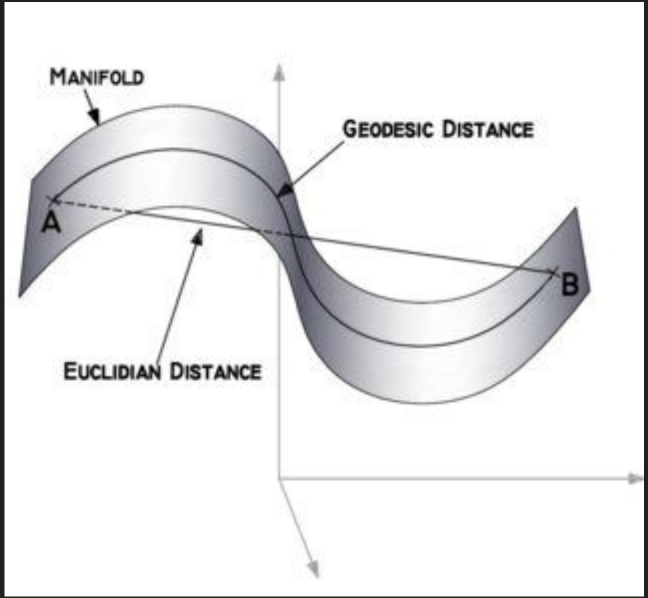


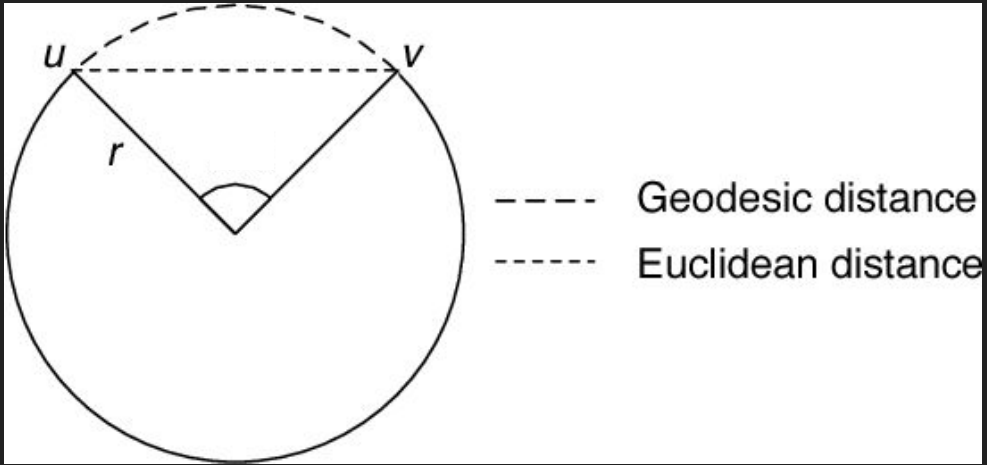
(a) Original Softmax Loss (b) Original Softmax Loss



(a) Softmax (b) ArcFace

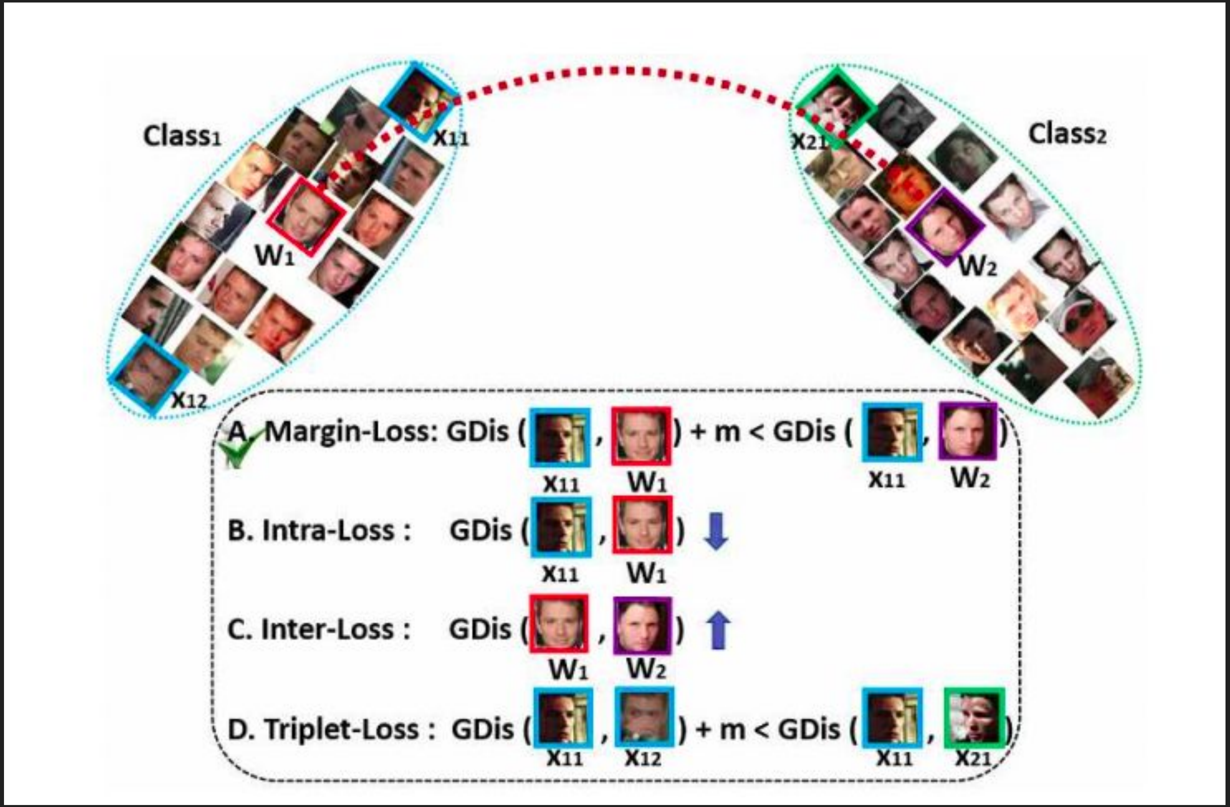
Euclidean Distance *vs* Geodesic Distance





ArcFace

Intuition



W_j: provides a kind of centre for each class.

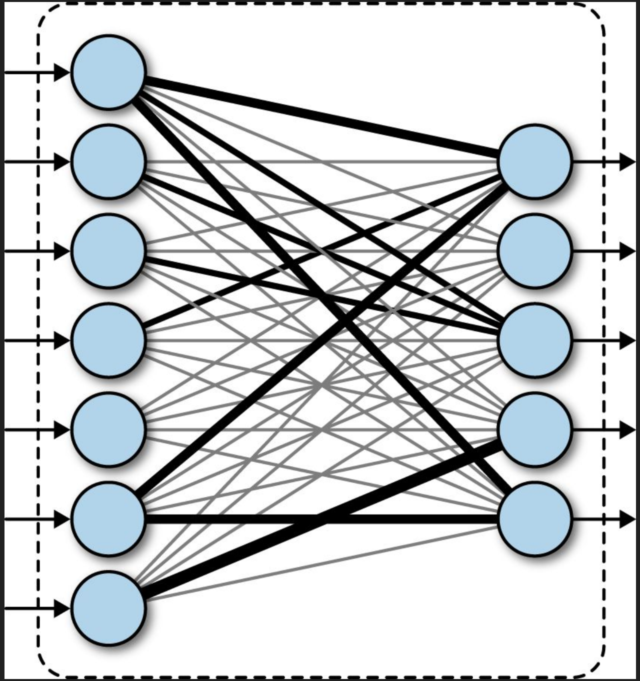
x_{jk}: samples of class j

GD(Intra-class) + m **<** GD(Inter-class)

Softmax Loss

$$L_1 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}},$$

where $x_i \in \mathbb{R}^d$ denotes the deep feature of the i-th sample, belonging to the y_i-th class. The embedding feature dimension d is set to 512 in this paper following [36, 43, 15, 35]. $W_j \in \mathbb{R}^d$ denotes the j-th column of the weight $W \in \mathbb{R}^{d\times n}$ and $b_j \in \mathbb{R}^n$ is the bias term. The batch size and the class number are N and n, respectively.



Formulation

$$L_1 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}},$$

$$L_2 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\cos\theta_{y_i}}}{e^{s\cos\theta_{y_i}} + \sum_{j=1, j\neq y_i}^{n} e^{s\cos\theta_j}}. \quad (2)$$

$$L_3 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))} + \sum_{j=1, j\neq y_i}^{n} e^{s\cos\theta_j}}. \quad (3)$$

- Set bias = 0
- $W^T x = |W|.|x|.\cos\theta$
- $|W| = 1$
- $|x| = 1$, and then scaled to s

So,

- $W^T x = s.\cos\theta$

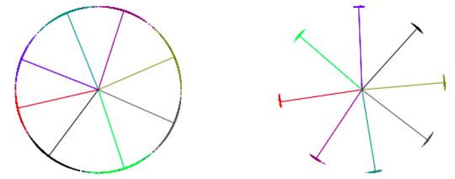
Formulation

$$L_1 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}},$$

$$L_2 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\cos\theta_{y_i}}}{e^{s\cos\theta_{y_i}} + \sum_{j=1,j\neq y_i}^{n} e^{s\cos\theta_j}}. \quad (2)$$

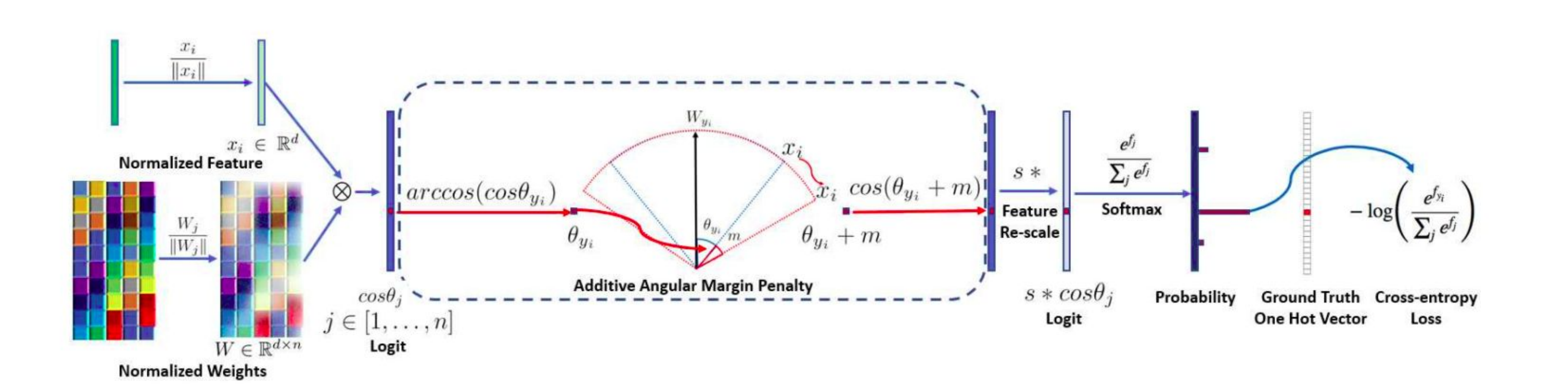
$$L_3 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))} + \sum_{j=1,j\neq y_i}^{n} e^{s\cos\theta_j}}. \quad (3)$$

- θ = arccos(cos θ)
- Add margin penalty m,
  - θ + m
- Take cosine,
  - cos (θ + m)



(a) Softmax (b) ArcFace

Training using ArcFace Loss



Other Losses

SphereFace, ArcFace, CosFace

- SphereFace: multiplicative angular margin
- ArcFace: additive angular margin
- CosFace: additive cosine margin

All enforce the intra-class compactness and inter-class diversity by penalising the target logit.

SphereFace

- Euclidean measurements not ideal for softmax due to its naturally angular distribution
- Also known as A-SoftMax (Angular Softmax)

$$L_{\text{ang}} = \frac{1}{N} \sum_i -\log \left(\frac{e^{\|\boldsymbol{x}_i\| \cos(m\theta_{y_i,i})}}{e^{\|\boldsymbol{x}_i\| \cos(m\theta_{y_i,i})} + \sum_{j \neq y_i} e^{\|\boldsymbol{x}_i\| \cos(\theta_{j,i})}} \right) \tag{6}$$

CosFace

$$L_{lmc} = \frac{1}{N}\sum_{i} -\log \frac{e^{s(\cos(\theta_{y_i,i})-m)}}{e^{s(\cos(\theta_{y_i,i})-m)} + \sum_{j\neq y_i} e^{s\cos(\theta_{j,i})}}, \tag{4}$$

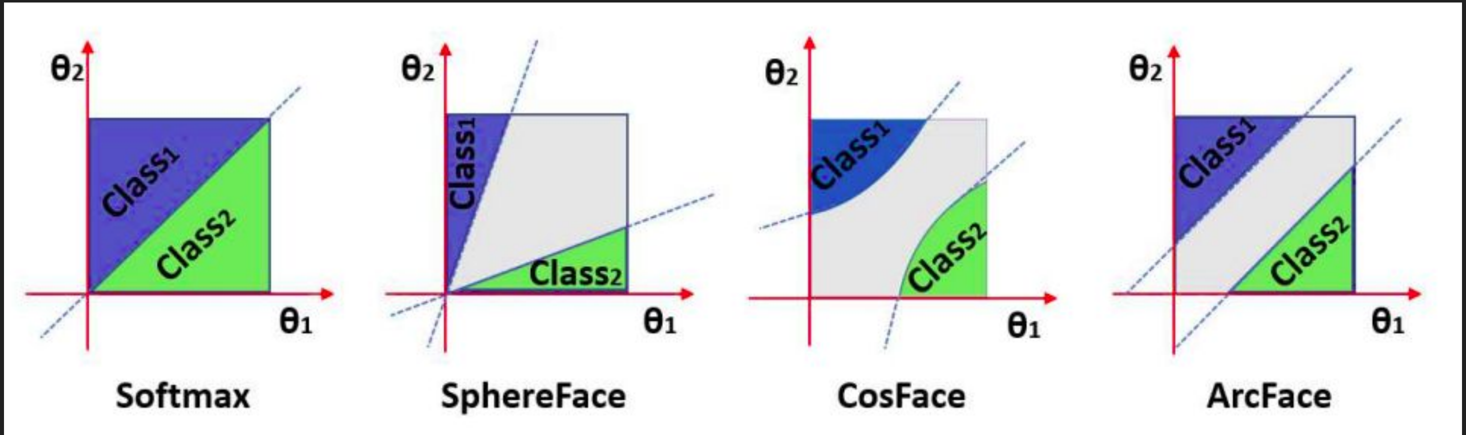
$$\begin{aligned} W &= \frac{W^*}{\|W^*\|}, \\ x &= \frac{x^*}{\|x^*\|}, \\ cos(\theta_j, i) &= {W_j}^T x_i, \end{aligned} \tag{5}$$

where N is the numer of training samples, x_i is the i-th feature vector corresponding to the ground-truth class of y_i, the W_j is the weight vector of the j-th class, and θ_j is the angle between W_j and x_i.

SphereFace, ArcFace, CosFace

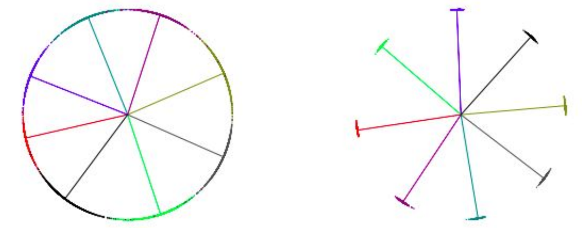
ArcFace has the exact correspondence to the geodesic distance.

The proposed ArcFace has a constant linear angular margin throughout the whole interval. By contrast, SphereFace and CosFace only have a nonlinear angular margin.



Advantages

- ArcFace directly optimises the geodesic distance margin by virtue of the exact correspondence between the angle and arc in the normalised hypersphere.



(a) Softmax (b) ArcFace

- ArcFace achieves state-of-the-art performance on ten face recognition benchmarks including large-scale image and video datasets.

Questions?