

Special Topics

MSDA 3440-01-F23

CLARK
UNIVERSITY



Project 1

Report on ArcFace

Submitted By:

Kabi Raj Tiruwa (C70292740)

Rahul Thapa Magar (C70293419)

Report on ArcFace: Additive Angular Margin Loss for Deep Face Recognition

In many facial recognition model the spread of features produced by its loss functions is quite large causing the boundaries between classes to be blurred this making it difficult for the model to correctly classify image ArcFace Additive Angular Margin helps reduced the spread by pushing together the images of the same individual while simultaneously pushing away image belonging to others according to the researchers this technique has helped them to outperform all other state of the art loss function which they applied on various datasets.

SoftMax is a common loss function used for facial recognition problems its works by imputing a vector of logits and normalizing them to be a probability distribution for each class this loss function calculates the distance between what the distribution of output should be and what the original distribution really is SoftMax doesn't enforces separation between classes which cause the classes to be closely clustered together.

Centre Loss tries to increase the disparity of the classes produces by SoftMax by calculating the center for each class and then moving its intra correlated features closer towards it that's creating class compactness. This is an improvement on SoftMax, but it comes with some drawbacks. First, calculating the center of classes is computationally expensive because it must calculate the distance of all features to find a center of each class. Secondly the center chosen isn't totally accurate because all features unable to be calculated ahead of time, so centers need to be created and redefined in each batch. Finally, the distance penalty that ap-plied to feature sis calculated using the Euclidean distance measurement which isn't the best way to separate SoftMax losses.

Triplet Loss is commonly used method that works by comparing three images, two images of same person and the third image of a different person the goal is to make the distance between the two images of the same person be significantly shorted than the distance the anchor in the imposter this is done by adding a margin clue to the positive image. The main problem with triplet loss is that it is very expensive there's a combinational explosion with large datasets as large number of images leads to an exponential number of parings. Additionally triple loss requires semi-hard sampling in order to learn effectively if the function would have compared two random images of say two different person chance are the distance between two images of one person would be much shorter that the distance to another person's image which would satisfy the loss function though it would teach the model much instead it is much more effective to compare two similar images

which makes the model work hard to adjust its weights and create a better classification this process of semi hard sampling is very computationally expensive.

Euclidean based margin is used mostly to separate features. SphereFace research discovered Euclidean measurements on ideal for SoftMax which has a naturally angular distribution sphere face utilizes SoftMax's natural angular distribution by imposing discriminative constraints on a hypersphere manifold allowing the inter and intra loss values to be controlled by a parameter M . This method is called angular SoftMax by constraining the weights and biases the new decision boundary only depends on θ_1 & θ_2 so its than just a matter of adding an integer m to control the decision boundary. m quantitatively controls the size of the angular margin simultaneously enlarging the inter-class margin and compressing the intraclass angular distribution so convenience of calculation and miss computed as an integer with a value greater than or equal to 1 if m equals 1 then the decision places of category 1 and category 2 are on the same plane if m is greater than or equal to 2 then there are two decision planes for category 1 and 2 and indicates that the maximum angle with the classification is smaller than the small angle of other classes by m times. A-SoftMax approximates the optimal value of m with its criteria being that the maximal inter-class distance should be smaller than the minimal intraclass distance there are two main drawbacks of A-SoftMax the first is due to the integer value of m which cause the curve at the target logit that is the logit which corresponds to the ground truth label To be very steep and thus hinders convergence secondly the decision margin of A-SoftMax depends on θ which leads to different margin for difference classes. As a result, in the decision space some interclass features have a larger margin while other have smaller margin which reduces its discriminating power.

CosFace adopts a different angular margin technique called large margin cosine loss or LMCL which aims to improve on sphere faces aforementioned shortcomings it does this by defining the decision margin and cosine space unlike sphere faces A-SoftMax loss which defines it in angular space it starts by reformulating the SoftMax loss by l_2 normalizing the features and weights to remove radial variance this addresses a-SoftMax's first issue of producing different margins for different classes which is the result of depending on the value of θ . As with A-SoftMax a margin value is added to increase the inter class distance and reduce the intra class distance. CosFace loss function maximizes $\cos \theta_1$ and minimizes $\cos \theta_2$ for c_1 to perform the large margin classification. This is superior to A-SoftMax's decision boundary whose margin is not consistent overall θ values making the decision boundary difficult to optimize.

ArcFace further improves the discriminative power achieved by SphereFace and CosFace by applying an additive angular margin loss unlike CosFace which applies an angular margin directly to the target logit. ArcFace applies it to the inverse of the angle using the arc cos function before using the cosine function to get back the target logit. It then rescales the logits by a fixed feature norm and rest is the same as the SoftMax loss function.

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

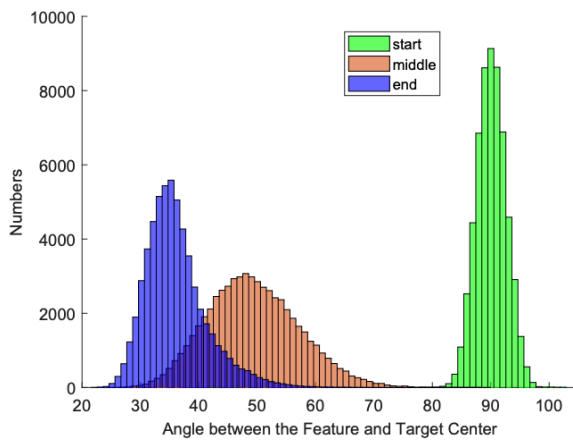
we start with the normal soft max loss function which is shown here for simplicity the bias is set to equal zero and the logit is transformed to be equal to the cosine distance after feature and weigh normalization. They old two normalized individual weight is said to equal 1 and the old two normalized embedding feature is rescale to as. The learned embedding features are thus distributed and hypersphere with a radius of s these nominalizations allowed the predictions to only depend on the angle between the feature and the weight. The embedding features are distributed around the feature center of the hyper sphere so we can add an additive angular margin m between each weight and featured and when adds the separation and compactness between clauses below is an example of class separation between ArcFace and SoftMax.

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

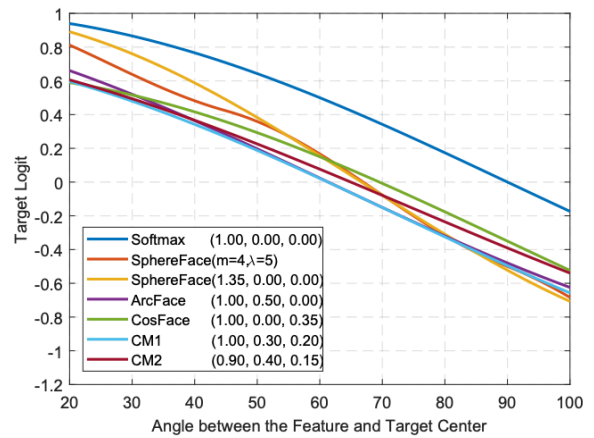
the experiment was made using images of eight different identities with around 1,500 images each as the image shows the SoftMax loss provides roughly separable feature embedding but produces noticeable ambiguity and decision boundaries while the proposed ArcFace loss can obviously enforce a more evident gap between the nearest classes ArcFace directly optimizes the geodesic distance margin by virtue of the exact correspondence between the angle and arc in the normalized hypersphere the chief state-

of-the-art performance on 10 face recognition benchmarks including live scale image and video data sets. It only needs several lines of code and it's extremely easy to implement in the computational graph based deep learning frameworks like PyTorch and TensorFlow furthermore ArcFace does not need to be combined with other loss functions in order to have stable performance and can easily converge on any training data sets it only adds negligible computational complexity during training current GPUs can easily support millions of identities for training and the model parallel strategy can easily support many more identities.

Arcface researchers also experimented by combining all the margins of SphereFace, ArcFace and CosFace in the United framework they referred to as CM.

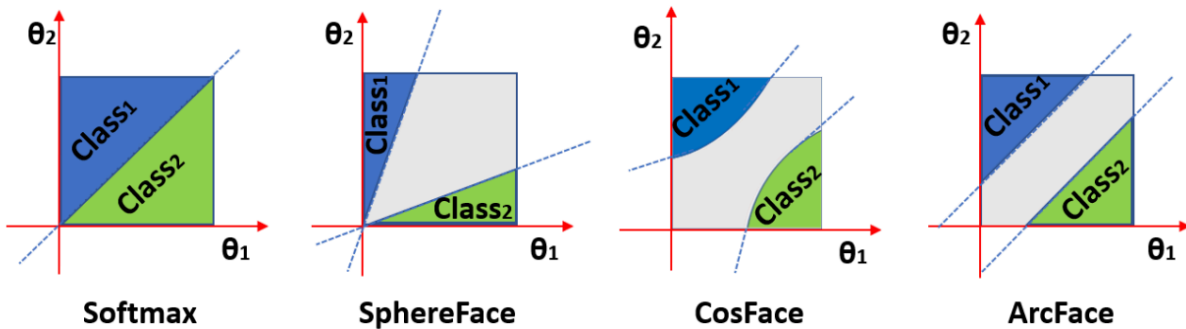


(a) θ_j Distributions



(b) Target Logits Curves

As the above figure shows combining all the margins also created target logit curves with a high-performance.







Above figure demonstrates decision boundaries for each loss function and a binary classification example. the dashed lines represent the boundary decision and the gray areas of the decision margins as you can see SoftMax doesn't create any decision margin

between classes whilst all the others due to some degree ArcFaces decision margin is best creating a constant linear angular margin throughout the whole interval this is due to ArcFaces additive angular margin having the exact correspondence to the geodesic distance. Sphereface and CosFace on the other hand when we have a nonlinear angular margin all of the models were trained using five data sets CASIA, VGGFace2, MS1MV2, MS1M-DeepGlint and Asian-DeepGlint each training set was separately employed in order to conduct fair comparison with other models during training the improvement from different settings was checked using several verification datasets besides the most widely used LFW and YTF data sets the performance of ArcFace was compared to recent large pose and large data sets CPLFW and CALFW finally the ArcFace model was tested on large scale image data sets as well as the video data set iQIYI-VID. Before testing the normalized face crops 112×112 were generated by utilizing 5 facial points ResNet50 and ResNet 100 CNN architectures were employed the embedded Network which is a widely used standard. The feature scale s is set to 64 and the angular margin m of the ArcFace is set to 0.5 which seems to be the best setting all experiments in the paper were implemented by the Apache MX net framework finally the batch size was set to 512 and was trained using four NVIDIA GPUs.

On CASIA the learning rate starts at 0.1 and is divided by 10 at 20K and 28K iteration and training process is finished at 32 K iteration.

On MS1MV2 the learning rate is divided at 100K, 160K iteration and finished at 180K iterations.

For testing the momentum is set to 0.9 and weight decay is 0.0005 only the feature embedding network is kept without the fully connected layer and the 512-D features are extracted for each normalized face. To get the embedding features for templates the feature centers of all images are calculated from the template or all the frames from the video sets finally all the overlap identities between the training set and the test set were removed for strict evaluations.

Loss Functions	LFW	CFP-FP	AgeDB-30
ArcFace (0.4)	99.53	95.41	94.98
ArcFace (0.45)	99.46	95.47	94.93
 ArcFace (0.5)	99.53	95.56	95.15
ArcFace (0.55)	99.41	95.32	95.05
SphereFace [17]	99.42	-	-
SphereFace (1.35)	99.11	94.38	91.70
CosFace [35]	99.33	-	-
CosFace (0.35)	99.51	95.44	94.56
 CM1 (1, 0.3, 0.2)	99.48	95.12	94.38
CM2 (0.9, 0.4, 0.15)	99.50	95.24	94.86
Softmax	99.08	94.39	92.33
Norm-Softmax (NS)	98.56	89.79	88.72
NS+Intra	98.75	93.81	90.92
NS+Inter	98.68	90.67	89.50
NS+Intra+Inter	98.73	94.00	91.41
 Triplet (0.35)	98.98	91.90	89.98
ArcFace+Intra	99.45	95.37	94.73
ArcFace+Inter	99.43	95.25	94.55
ArcFace+Intra+Inter	99.43	95.42	95.10
 ArcFace+Triplet	99.50	95.51	94.40

As the red arrow points out ArcFace perform best with all three validation sets the ArcFace, CosFace and SphereFace hybrids referred to as CM1 and CM2 also perform very well better than SphereFace alone in both settings. Also, interestingly the ArcFace triplet the hybrid loss performed better than the triplet lost by itself as indicated by the blue arrows and of course SoftMax at the worst performance of all due to its lack of intraclass compactness and interclass disparity.

LFW and YTF datasets of the most widely used benchmark for unconstrained face verification on images and videos the ArcFace researcher followed the unrestricted would label outside data protocol to observe its performance as you can see in the table, they trained ArcFace on MS1MV2 with ResNet100 which beat all the other losses by a significant margin on both verification sets. The rest of the paper just discusses ArcFace performance with all the other verification and test sets all of which ArcFace outperform the other losses.