MSDA-3100-01 Applied Deep Learning
Project 1
ArcFace Loss Implementation for Multi-Class Classification

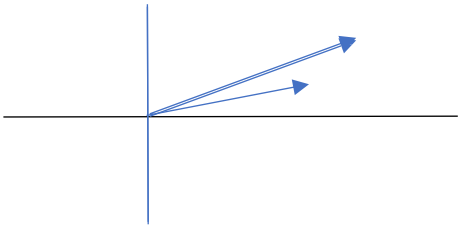
In 2019, a famous research paper entitled "ArcFace: Additive Angular Margin Loss for Deep Face Recognition" introduced the idea of the ArcFace loss function and to this date, it has the highest face recognition accuracy on some of the popular face recognition benchmarks.

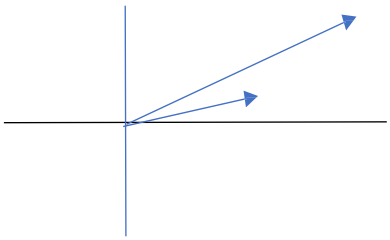
The fundamental idea in ArcFace is to change the similarity measure from a Euclidean distance (L2 Norm) to a Cosine distance. The motivation is that cosine distance is somewhat invariant to partial noise in the data (especially in high dimensional cases.

For example, take a look at the 2-D case where we have two points represented by vectors $\begin{bmatrix}5\\2\end{bmatrix}$, and $\begin{bmatrix}7\\4\end{bmatrix}$ as shown below. The Euclidean distance between these two points 2.83, while the angle between these vectors can be computed as :

The dot product between two vectors is: $A.B = \|A\|\|B\| \, Cos\, \theta$

$$\theta = ArcCos\left(\frac{A.B}{\|A\|\|B\|}\right) = ArcCos\left(\frac{35+8}{5.38x8.06}\right) = ArcCos(0.99) = 7.8\ degrees$$



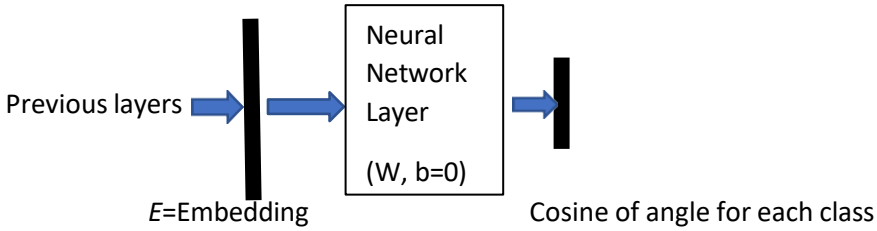


If the second point increases in its individual feature by 50% and changes to $\begin{bmatrix}10.5\\6\end{bmatrix}$, the Euclidean distance will increase from 2.83 to 6.8, however if you compute the new angle between $\begin{bmatrix}5\\2\end{bmatrix}$, and $\begin{bmatrix}10.5\\6\end{bmatrix}$ it remains the same.

Thus, the main motivation in the ArcFace loss function is to use the cosine distance as the similarity measure between two vectors. If the angle between the vectors is small, the similarity is higher and vice versa.

From a neural network point of view, if the input to a network is a vector, and the embedding produced by the network as it multiplies the input with the weights is also a vector, then there will be an angle produced between the input vector and the embedding vector. Thus a neural network can be thought of as capable of rotating the input vector in the angle space.

If the embedding vector is already produced by a network, and we needed to determine the cosines of the angles for each output class, we can pass it through a simple linear network as shown below.



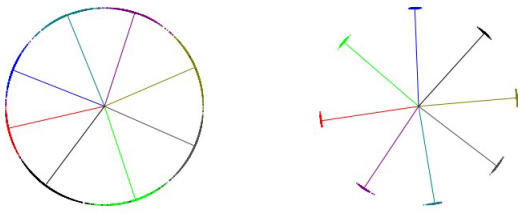
where E = Embedding, then output of network in above picture is $W^T E$

For example, if the Embedding size is 100x1, then the *W* matrix will be 100x10 for a 10-class problem. You can think of each column of *W* as a 100x1 vector which when multiplied with the Embedding will provide a Cosine of the angle between the embedding and that column.

$E.E_c = \|E\|\|E_c\|\text{Cos}\theta$

If E and E_c are normalized to have unit magnitude i.e., $\|E\| = 1$, $\|E_c\| = 1$, then $E.E_c = \text{Cos}\theta$

ArcLoss uses an additional simple network (on top a regular CNN that produces the embedding) as shown above to output the cosines of angles such that clusters of classes are created in the embedding space that are farther away from each other in the angle space. It also compresses the intra-class distance while maximizing the inter-class distance as shown below.



(a) Softmax (b) ArcFace

The ArcFace loss function is simply a softmax implementation of the rotated embedding space as shown below. The s is the scale in the following equation so that the hypersphere can be made bigger to create a good separation between the classes when the number of classes is large.

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

The creators of the ArcLoss function also introduced a parameter called margin (similar to margin in the Siamese contrastive loss function) which tries to make sure that the classes are separated by at least this margin amount in the angle space. The final loss function appears as:

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

1) Read carefully the paper titled:

J. Deng, J. Guo, N. Xue and S. Zafeiriou, "ArcFace: Additive Angular Margin Loss for Deep Face Recognition," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, 2019, pp. 4685-4694, doi: 10.1109/CVPR.2019.00482.

2) Write a report that explains your understanding of the paper's presented technique.
3) Write a python program using Pytorch library to build the above loss function for classification of MNIST digits and CIFAR10 images.
4) Your submission should be a zip file (named as lastname_firstname_ID_Project1.zip, e.g. mark_philip_0123456_Project_1.zip) that includes:
    a. All written python files.
    b. All generated figures as png files.
    c. A PDF report that describes your codes and results (with screenshots) including a conclusion section about your achieved results.