# ArcFace: Additive Angular Margin Loss for Deep Face Recognition

Jiankang Deng；Jia Guo；Stefanos Zafeiriou

帝国理工学院【英】；DeepInSight公司【中】

## 论文简介

这篇论文原名是ArcFace（additive angular margin），但是由于与虹软重名，后改名为Insight Face。是2018年1月发表的，是基于AMsoftmax的改进，主要提出了两个创新点：

1）角度距离比余弦距离在对角度的影响方面更加直接，故将（cosθ-m）的形式改成了在cos（θ+m）的形式。

2）作者提出了一种称为IR的网络结构，本质对Resnet的block进行了一些改进，文章说它更适合对人脸图片的训练。

个人理解：在本文之前，已经有了angular margin乘法的算法（Sphereface）和cosine margin加法的算法（AMsoftmax），本文提出angular margin加法的算法算是亦步亦趋。

开源地址：https://github.com/deepinsight/insightface

## 相关研究

既然本篇论文是基于AMsoftmax的改进，那么不妨先看一下AMsoftmax（Additive Margin Softmax）。

### 2.1 AMsoftmax的简述

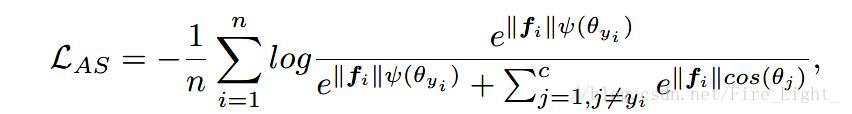
AMsoftmax是在论文《Additive Margin Softmax for Face Verification》中提出来的一种用于人脸识别的损失函数，发表的时间也是2018年1月，而作者中包括了A-Softmax和Norm Face的一作。

在目前的大多数研究中，一般是将人脸验证任务视为度量学习问题，因此学习类内间距较小，类间差异较大的人脸特征对于实现良好的性能具有重要意义。之前的L-Softmax和A-Softmax引入了角间距的概念，用于改进传统的softmax loss函数，使得人脸特征具有更大的类间距和更小的类内距。作者在这些方法的启发下，提出了一种更直观和更易解释的additive margin Softmax (AM-Softmax)。同时，本文强调和讨论了特征正则化的重要性。

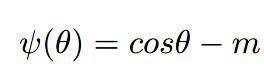
### 2.2 AMsoftmax的分析

AMsoftmax是一种对sphereface的改进版本，具体的方法是把角度裕量改成了余弦裕量，即从cos(mθ)改进为cos(θ)+m，主要的好处是这样改进之后容易收敛。

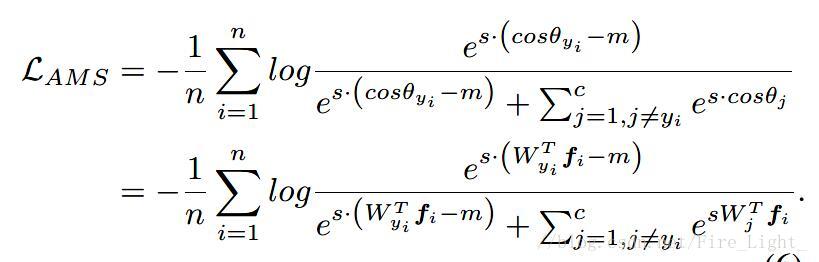
1）在Sphereface中提出的Asoftmax损失函数公式：

20180318170439510

2）AMsoftmax中提出的损失函数如下：



与ASoftmax中定的的类似，可以达到减小对应标签项的概率，增大损失的效果。然后根据Normface，对f进行归一化，乘上缩放系数s，最终的损失函数变为：



3）Asoftmax与AMsoftmax的比较

作者认为，Asoftmax是用m乘以θ，是角度距离，计算loss时是进行乘法操作；而AMSoftmax是用cosθ减去m，是余弦距离（可以把m理解成为cosα），计算loss时是进行加法操作。

使用传统的Softmax的时候，角度距离和余弦距离是等价的，但是进行决策边界优化的时候，角度距离和余弦距离就有所不同了。具体来说，在角度处于0或者pi的附近时，余弦距离相对更密集（区别性更小），故优化角度距离比优化余弦距离更有效果。

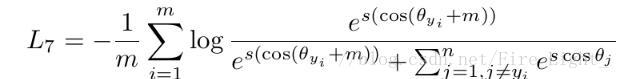
那么为什么AMsoftmax选择的是优化余弦距离呢？原因就是优化角度距离时会涉及到arccos操作（W和f的内积之后进行arccos运算），计算量过大。考虑到计算量上的花费，优化余弦距离能以更少的计算开销达到更高的性能。

sphereface中只对w进行了归一化；AMsoftmax对w和x都进行了归一化处理。

## 算法解析

### 损失函数的修改

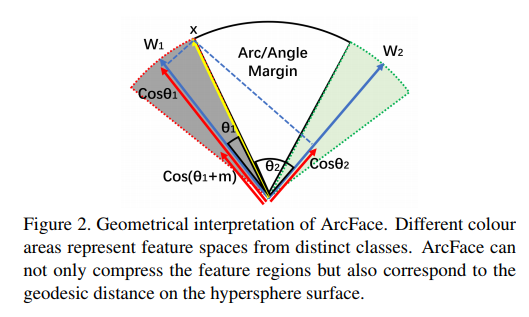
Arcface中提出的损失函数如下：



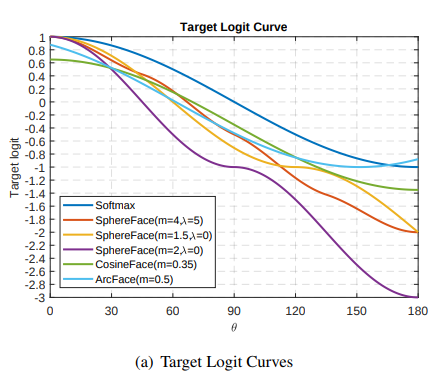
在加了裕量（在角度值较小的时候很有用），相当于原始的余弦函数沿x轴左移。这样做了之后，在角度值为pi的时候分类难度基本不变，而角度值较小的时候分类更加困难。该损失函数是有几何解释的，角度空间中的裕量对应于超球流形上的弧距离。

从公式上来看，该损失函数与AMsoftmax损失函数唯一的不同点就是Arcface将AMsoftmax中的常量m提到了cos中去。之所以这样做的原因是源于AMsoftmax的论文中得到一个论断，即角度距离比余弦距离在对角度的影响更加直接。

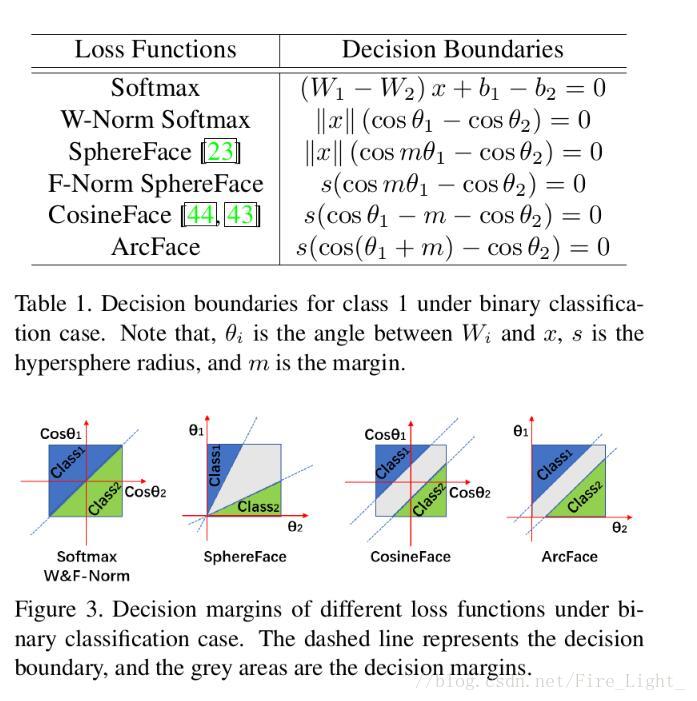
下图是Arcface、SphereFace和 CosineFace的比较，红色的线代表测地线距离。从图中可以看出来：1）Arcface可以使θ1和θ2的测地线距离大小类似。2）如果使用（cosθ-m）的形式，则测地线一定有一部分会在扇形区域之外；而使用cos（θ+m）的形式，则测地线一定在[0,R]内。

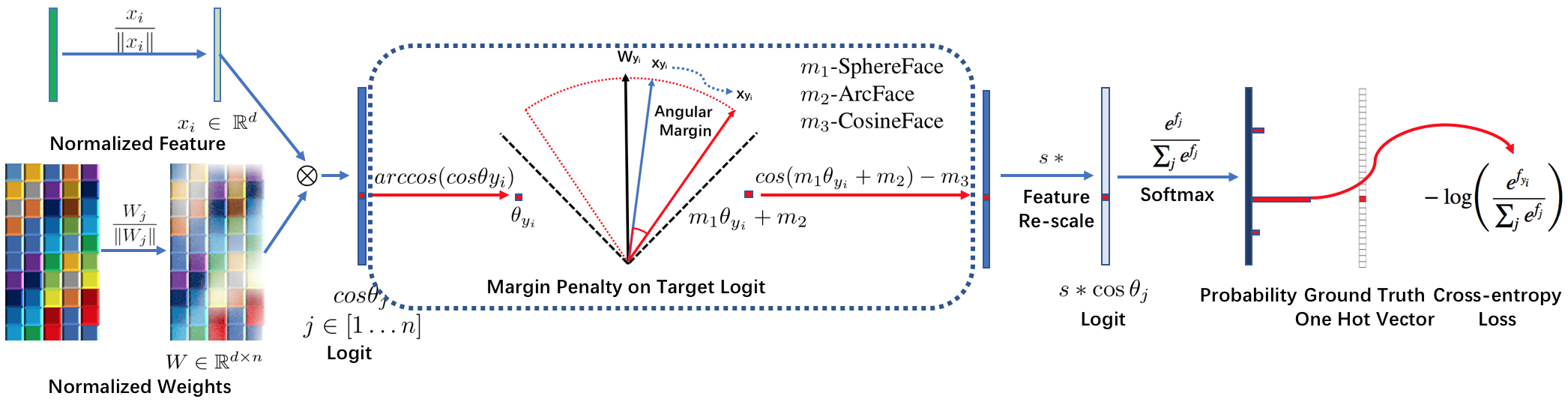


相应的logit曲线（fc的输出向量）:



比较一下softmax系列的各种loss：



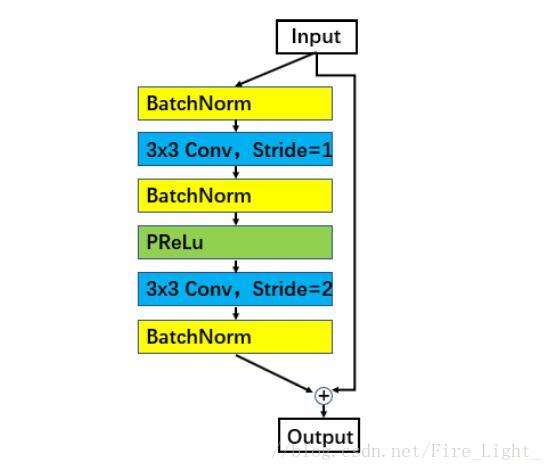


### 网络设置

1）本篇论文尝试了很多网络和不同超参数设置，包括了MobileNet、DenseNet、SENet、Inception-Resnet-V2和Dual path Network (DPN)，进行了训练时间和精度的权衡，这里面有很多超参数设置可以参考。

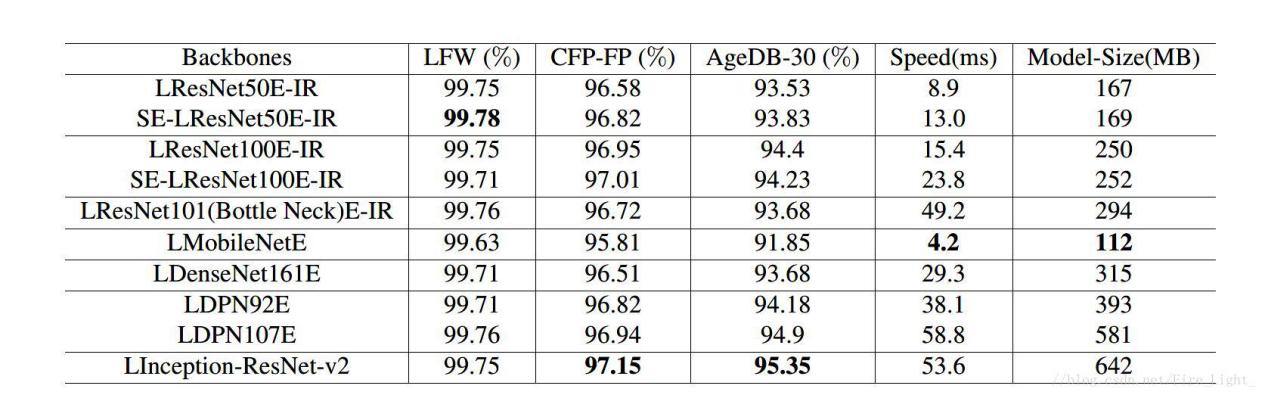
2）尝试了训练完之后，使用triplet-loss进行fineturn，能提高一定的精度。

3）提出了名为IR的网络组件，如下：

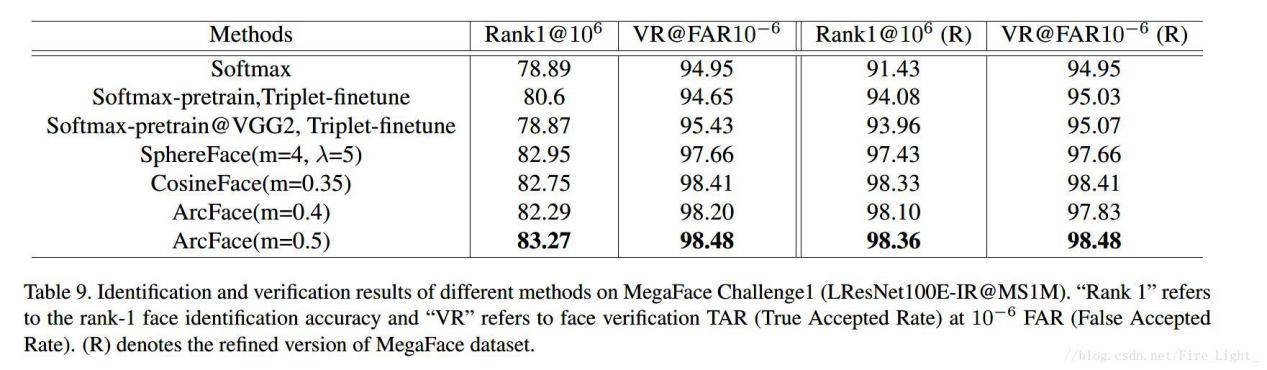


## 实验与结论

1）在lfw的实验



2）在MegaFace的实验结果：



该算法刚发表的时候是MegaFace榜第一，达到了98.36%的成绩，但是因为作者对动了MegaFace中的数据，将FaceScrub与MegaFace1M干扰项中重叠的部分去除了，（这也就是表中R的含义），因此不能体现该算法对其他算法有绝对的优势。

现在，Arcface已经排不进MegaFace榜的前十，megaface数据集成为了各个算法刷榜的数据集；而LFW基本已经淘汰(?)。

**参考资料**

<https://blog.csdn.net/fire_light_/article/details/79602705>

https://blog.csdn.net/u014230646/article/details/79487720