人脸识别之 ArcFace

论文：ArcFace: Additive Angular Margin Loss for Deep Face Recognition

首先我们需要来对比一下ArcFace，AM-softmax，A-softmax和softmax之间的联系和区别。可以这样认为, A-softmax是在传统softmax的基础上，通过m和角度相乘，引进了角度间隔（angular margin) m; AM-softmax是对A-softmax的改进, 把m由cos函数内引到cos函数外，将乘法变成加法运算，为余弦间隔；ArcFace则是在cos函数里加上一个角度间隔（angular margin）m。有意思，呵呵呵😊😊😊这四种损失函数如表格1所示：

表格1：四种softmax损失函数对比

损失函数	公式	备注
softmax	$L_1 = -\frac{1}{m}\sum_{i=1}^{m}\log\frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}}$	
A-softmax	$L_3 = -\frac{1}{m}\sum_{i=1}^{m}\log\frac{e^{\|x_i\|\cos(m\theta_{y_i})}}{e^{\|x_i\|\cos(m\theta_{y_i})} + \sum_{j=1, j\neq y_i}^{n} e^{\|x_i\|\cos\theta_j}}$	对参数L2正则化后引入角度间隔m
Am-softmax	$L_6 = -\frac{1}{m}\sum_{i=1}^{m}\log\frac{e^{s(\cos(\theta_{y_i})-m)}}{e^{s(\cos(\theta_{y_i})-m)} + \sum_{j=1, j\neq y_i}^{n} e^{s\cos\theta_j}}$	对特征和参数L2正则化后，引入余弦间隔
ArcfFace	$L_7 = -\frac{1}{m}\sum_{i=1}^{m}\log\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))} + \sum_{j=1, j\neq y_i}^{n} e^{s\cos\theta_j}}$	对特征和参数L2正则化后，在con函数里引入角度间隔m

ArcFace的几何表示如下图所示：在二分类情况下，对于类别1，ArcFace的边界决策函数为 $s(cos(\theta_1 + m) - cos(\theta_2)) = 0$

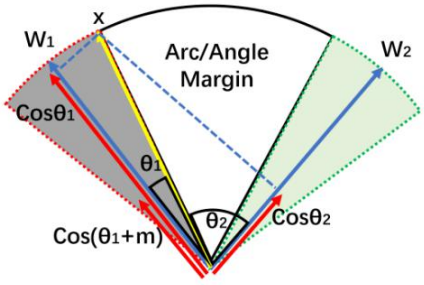


Figure 2. Geometrical interpretation of ArcFace. Different colour areas represent feature spaces from distinct classes. ArcFace can not only compress the feature regions but also correspond to the geodesic distance on the hypersphere surface.

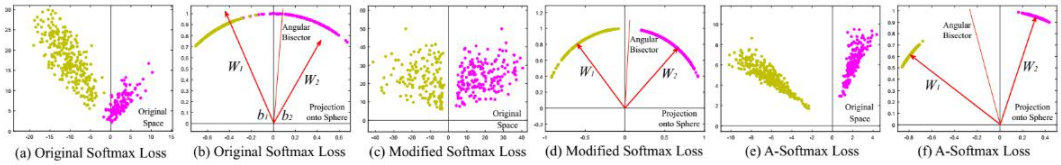
A-softmax

该论文主要是解决Open-set数据集上的人脸识别任务。理想的Open-set人脸识别学习到的特征应当在特定的度量空间中，满足同一类的最大类内距离小于不同类的最小类间距离。然而softmax loss仅仅能够使得特征可分，还不能够使得特征具有可判别性。尽管有一些方法通过结合softmax loss和 contrastive loss，center loss去提高特征的可判别性，但是contrastive loss和center loss需要精心地构建图像对和三元组，不仅耗时而且构建的训练对会对识别性能影响很大。因此，在这篇论文中，作者提出了angular softmax（A-Softmax）loss。

对于一个二分类的softmax的决策边界是$(\mathbf{W}_1-\mathbf{W}_2)\mathbf{x}+b_1-b_2=0$，如果限制$\|\mathbf{W}_1\|=\|\mathbf{W}_2\|=1\ and\ b_1=b_2=0$，那么边界决策函数就就可以变形为$\|\mathbf{x}\|\left(cos\left(\theta_1\right)-cos\left(\theta_2\right)\right)=0$，其中$\theta_i$是权重向量$\mathbf{W}_i$和特征向量$\mathbf{x}$之间的夹角，那么这个边界决策函数仅仅由$\theta_1$和$\theta_2$所决定，因此损失函数关注的是特征的角度可分性。在此基础上，作者又引进了一个整数变量m（m>=1），m控制着角度间隔，对类别1和类别2的边界决策函数变形为$\|\mathbf{x}\|\left(cos\left(m\theta_1\right)-cos\left(\theta_2\right)\right)=0$和$\|\mathbf{x}\|\left(cos\left(\theta_1\right)-cos\left(m\theta_2\right)\right)=0$。这就是二分类的A-Softmax loss。Table 1罗列了softmax Loss，modified softmax loss和A-softmax loss 在二分类的情况下的边界决策函数的形式，Figure 2 则是效果展示。虽然是二分类形式，但是也可以一般化成多类别的分类。

Loss Function	Decision Boundary
Softmax Loss	$(\boldsymbol{W}_1-\boldsymbol{W}_2)\boldsymbol{x}+b_1-b_2=0$
Modified Softmax Loss	$\|\boldsymbol{x}\|(\cos\theta_1-\cos\theta_2)=0$
A-Softmax Loss	$\|\boldsymbol{x}\|(\cos m\theta_1-\cos\theta_2)=0$ for class 1 $\|\boldsymbol{x}\|(\cos\theta_1-\cos m\theta_2)=0$ for class 2

Table 1: Comparison of decision boundaries in binary case. Note that, θ_i is the angle between $\boldsymbol{W}_i$ and $\boldsymbol{x}$.



(a) Original Softmax Loss (b) Original Softmax Loss (c) Modified Softmax Loss (d) Modified Softmax Loss (e) A-Softmax Loss (f) A-Softmax Loss

Figure 2: Comparison among softmax loss, modified softmax loss and A-Softmax loss. In this toy experiment, we construct a CNN to learn 2-D features on a subset of the CASIA face dataset. In specific, we set the output dimension of FC1 layer as 2 and visualize the learned features. Yellow dots represent the first class face features, while purple dots represent the second class face features. One can see that features learned by the original softmax loss can not be classified simply via angles, while modified softmax loss can. Our A-Softmax loss can further increase the angular margin of learned features.

上面以二分类的情况作为一个例子，下面就开始描述一般化的形式。

对于输入特征$\mathbf{x}_i$和它的标签y_i，softmax loss的一般化函数可表示如下

$$L=\frac{1}{N}\sum_i L_i=\frac{1}{N}\sum_i-\log\left(\frac{e^{f_{y_i}}}{\sum_j e^{f_j}}\right)$$

其中N表示训练样本的次数，卷及网络中，f通常是全连接层W的输出，所以$f_j=\boldsymbol{W}_j^T\boldsymbol{x}_i+b_j$，将f函数带入，可以将公式变换为如下：

$$\begin{aligned}L_i&=-\log\left(\frac{e^{\boldsymbol{W}_{y_i}^T\boldsymbol{x}_i+b_{y_i}}}{\sum_j e^{\boldsymbol{W}_j^T\boldsymbol{x}_i+b_j}}\right)\\&=-\log\left(\frac{e^{\|\boldsymbol{W}_{y_i}\|\|\boldsymbol{x}_i\|\cos(\theta_{y_i,i})+b_{y_i}}}{\sum_j e^{\|\boldsymbol{W}_j\|\|\boldsymbol{x}_i\|\cos(\theta_{j,i})+b_j}}\right)\end{aligned}$$

虽然可以通过modified softmax loss学习特征，但是这些特征不一定可判别。为了进一步将强特征的可判别性，作者将角度距离加入到损失函数中，即A-Softmax loss，一般化的公式如下：

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

其中，$\theta_{y_i,i}$的范围为$\left[0, \frac{\pi}{m}\right]$，为了消除该限制，并能够在CNN中最优，定义角度函数$\psi(\theta_{y_i,i})$扩展$cos(\theta_{y_i,i})$的定义范围，这个函数在$\left[0, \frac{\pi}{m}\right]$区间内等价于$cos(\theta_{y_i,i})$，因此A-softmax的公式可表示为：

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

其中$\psi(\theta_{y_i,i}) = (-1)^k\cos(m\theta_{y_i,i}) - 2k$，$\theta_{y_i,i} \in \left[\frac{k\pi}{m}, \frac{(k+1)\pi}{m}\right]$ and $k \in [0, m-1]$。可以看出，当m=1时，A-softmax loss就变成modified softmax loss。当m增大时，角度距离也会增加，当m=1时，角度距离为0. 从决策函数角度解释A-softmax loss，A-softmax loss 对于不同的类别采用了不同的决策函数，每个决策函数比原始的更加严格，因而产生了角度距离。

总的来说，论文中的A-softmax loss并不是新的loss，只是在原始的softmax loss的基础上做了改进，加入整数m，产生了角度距离，结果表明这个角度距离可以使得特征具有可判别性，而且m越大，可判别性就越大。

AM-softmax

论文：Additive Margin Softmax for Face Verification

在此篇论文中，作者提出了新的损失函数，AM_softmax。嗯，我们可以把AM-softmax看做是对A-softmax损失函数的改进，两者是很相似的，我刚写过A-softmax的博客，详情请点击这里。两者最大的不同之处在于：A-softmax是用margin m与θ相乘，而AM-softmax的margin则是$cos(\theta)-m$，一个是角度距离(angular margin)，一个是余弦距离(cosine margine)。当用传统的softmax作为损失函数的时候，角度距离和余弦距离是等价的，即：$cos(\theta_1)=cos(\theta_2)\Rightarrow\theta_1=\theta_2$，但是当试着去推动边界的时候，余弦距离和角度距离就不再等价了。

回顾一下A-softmax损失函数，公式表示如下：

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

其中，$\psi(\theta_{y_i,i})=(-1)^k\cos(m\theta_{y_i,i})-2k$，$\theta_{y_i,i}\in[\frac{k\pi}{m},\frac{(k+1)\pi}{m}]$ and $k\in[0,m-1]$，m通常是一个大于1的整数，λ是一个表示推动分类边界难度的超参数。相比较而言，AM-softmax定义$\psi(\theta)$更加简单和直观，$\psi(\theta)=cos(\theta)-m$。为了提高收敛速度，作者又引进一个超参数s，所以AM-softmax的最终形式为：

$$\begin{aligned}\mathcal{L}_{AMS}&=-\frac{1}{n}\sum_{i=1}^{n}log\frac{e^{s\cdot(cos\theta_{y_i}-m)}}{e^{s\cdot(cos\theta_{y_i}-m)}+\sum_{j=1,j\neq y_i}^{c}e^{s\cdot cos\theta_j}}\\&=-\frac{1}{n}\sum_{i=1}^{n}log\frac{e^{s\cdot(W_{y_i}^T\boldsymbol{f}_i-m)}}{e^{s\cdot(W_{y_i}^T\boldsymbol{f}_i-m)}+\sum_{j=1,j\neq y_i}^{c}e^{sW_j^T\boldsymbol{f}_i}}.\end{aligned}$$

作者对additive margin做了一个直观的分析，用了一个二维特征作为例子，如图1所示，对一个具有二维的特征，正则化后，特征被表示在一个圆中，传统softmax的决策边界即是向量P_0，那么$W_1^TP_0=W_2^TP_0$；而AM-softmax是以决策区域替代决策边界，对于类别1的边界为向量P_1，定义$W_1^TP_1-m=W_2^TP_2$，那么$m=(W_1-W_2)^TP_1=cos(\theta_{W_1},P_1)-cos(\theta_{W_2},P_1)$。更进一步假设所有的类别都具有相同的方差，$P_2$是类别2的边界向量，那么$cos(\theta_{W_2},P_1)=cos(\theta_{W_1},P_2)$，所以$m=cos(\theta_{W_1},P_1)-cos(\theta_{W_1},P_2)$

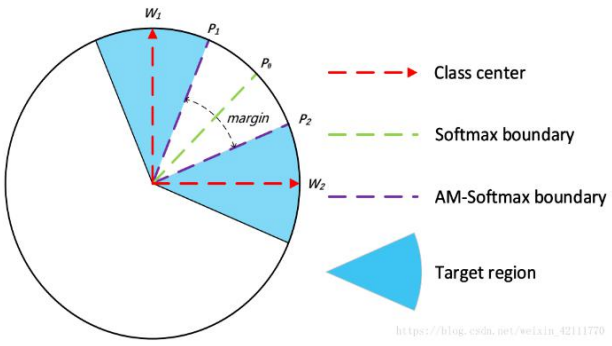


图1：传统softmax的决策边界和AM-softmax的决策边界

作者还提出，特征是否正则化处理还取决于图片的质量，高质量的图片提取出来的特征范数比较大，低质量的图片提取出来的特征范数小，那么特征正则化（feature normalization）后，在后向传播的时候，低质量的图片特征会产生更大的梯度，也会获得网络更多的注意力，如下图所示，因此，对于低质量图片的数据集，特征正则化是最适合的。也因此设置s=30。