人脸识别中Softmax-based Loss的演化史

加入极市专业CV交流群，与**6000+来自腾讯，华为，百度，北大，清华，中科院**等名企名校视觉开发者互动交流！更有机会与**李开复老师**等大牛群内互动!
同时提供每月大咖直播分享、真实项目需求对接、干货资讯汇总，行业技术交流。关注 **极市平台** 公众号 ，回复 **加群**，立刻申请入群~

近期，人脸识别研究领域的主要进展之一集中在了Softmax Loss的改进之上;在本文中，旷视研究院（上海）（MEGVII Research Shanghai）从两种主要的改进方式 - 做归一化以及增加类间margin--展开梳理，介绍了近年来基于Softmax的Loss的研究进展。

目录

引言

关于人脸识别领域Softmax Loss相关的科普文章其实很多。例如[人脸识别的LOSS（上，下）]以及[人脸识别最前沿在研究什么？]等文章分别从纸和目前主流工作的角度做了梳理。因此，本文不再挨个盘点时下各个纸所做的工作，而是从人脸识别中的Softmax Loss的历史发展脉络这个角度出发，沿着这条时间线详细介绍Softmax Loss的各种改进在当时的背景下是如何提出来的。

Softmax简介

Softmax Loss因为其易于优化，收敛快等特性被广泛应用于图像分类领域。然而，直接使用softmax loss训练得到的特征拿到检索，验证等“需要设阈值”的任务时，往往并不够好。

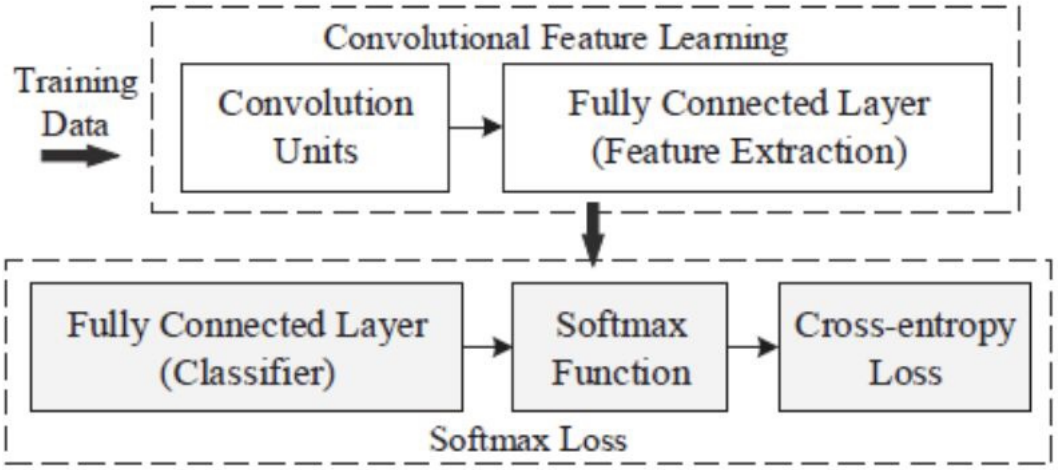
这其中的原因还得从Softmax的本身的定义说起，Softmax loss在形式上是softmax函数加上交叉熵损失，它的目的是让所有的类别在概率空间具有最大的对数似然，也就是保证所有的类别都能分类正确，而retievel和验证任务所需要的是一个泛化性能更好的度量空间（公制空间）。保证分类正确和保证一个泛化性优良的度量空间这两者之间虽然相关性很强，但并不直接等价。

因此，近年来，面部识别领域的主要技术进展集中在如何改进softmax的损失，使得既能充分利用其易于优化，收敛快的优良性质，又使得其能优化出一个具有优良泛化性的度空间。而这些技术改进主要又能被归为两大类别，做归一化以及加缘。以下从这两个方面进行一些梳理。

首先从一个简单的基于Softmax Loss的例子出发。下图描述了基于softmax loss做分类问题的流程。输入一个训练样本，倒数第二层的特征提取层输出特征x，和最后一层的分类层的类别权重矩阵

$$\mathbf{W} = \{\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_K\}$$

相乘，得到各类别的分数，再经过softmax函数得到normalize后的类别概率，再得到交叉熵损失。



图片来自[FaceRecSurvey]

类别重量

$\mathbf{w}_k$

可看作是一个类别所有样本的代表。

f_i^k

是样本feature和类别weight的点积，可以认为是样本和类别的相似度或者分数。通常这个分数被称为logit。

Softmax能够放大微小的类别间的logit差异，这使得它对这些微小的变化非常敏感，这往往对优化过程非常有利。我们用一个简单的三分类问题以及几个数值的小实验来说明这个问题。假设正确的类别为1.如下表的情况（d）所示，正确类别的概率才1/2，并不高。

如果要进一步提高正确类别概率，需要正确类别分数远高于其他类别分数，需要网络对于不同样本（类别）的输出差异巨大。网络要学习到这样的输出很困难。然而，加了softmax操作之后，正确类别概率轻松变为

$$\frac{e^{10}}{e^{10}+e^5+e^5} = 98.7\%$$

，已经足够好了。

情况	(f^1, f^2, f^3)	$p^i = \frac{f^i}{\sum_j f^j}$	$p^i = \frac{e^{f^i}}{\sum_j e^{f^j}}$
a	(1, 1, 1)	(33%, 33%, 33%)	(33%, 33%, 33%)
b	(2, 1, 1)	(50%, 25%, 25%)	(57.6%, 21.2%, 21.2%)
c	(5, 1, 1)	(71.43%, 14.29%, 14.29%)	(96.5%, 1.77%, 1.77%)
d	(10, 5, 5)	(50%, 25%, 25%)	(98.7%, 0.66%, 0.66%)

可见，softmax中的指数操作，可以迅速放大原始的logit之间的差异，使得“正确类别概率接近于1”的目标变得简单很多。这种效应可以称为“强者通吃”。

归一化（归一化）

归一化（归一化），是人脸识别领域中一个重要的方法它的做法实际上非常简单归一化的定义如下。：

- 对功能做归化（feature normalization）的定义为

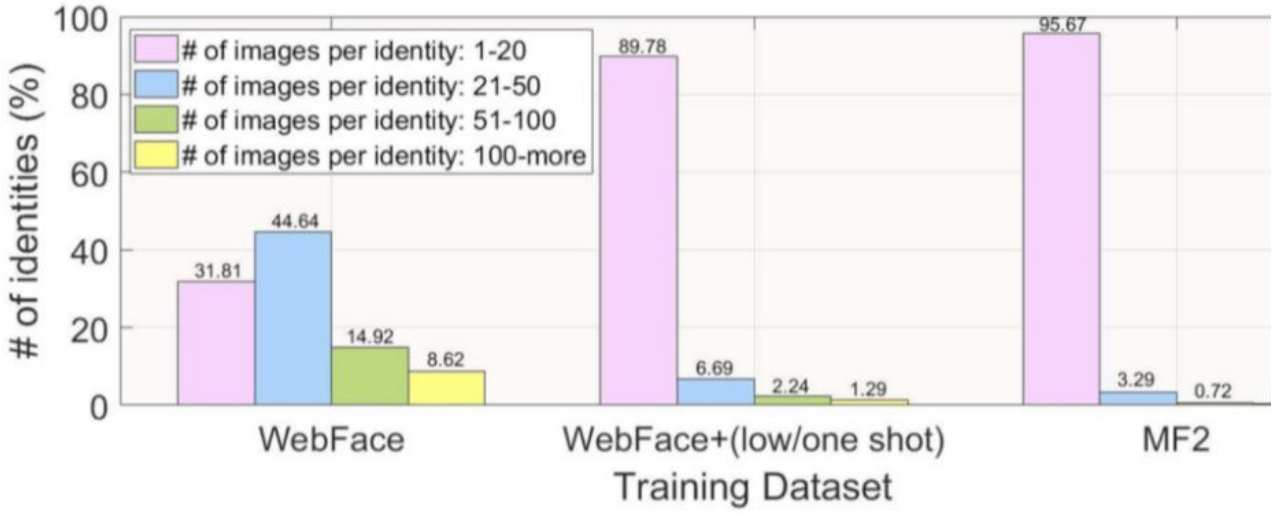
$$\mathbf{x} \rightarrow \frac{\mathbf{x}}{||\mathbf{x}||_2}$$

- 对重做一归化（weight normalization）的定义为

$$\mathbf{w_k} \rightarrow \frac{\mathbf{w_k}}{||\mathbf{w_k}||_2}$$

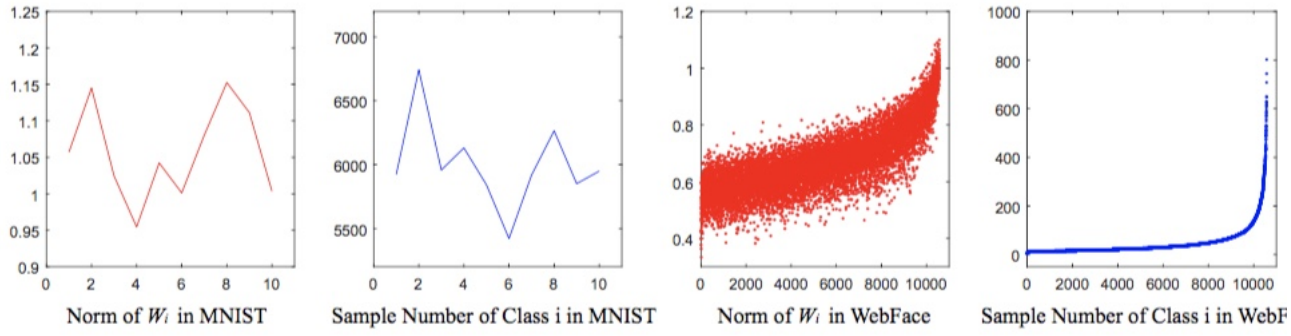
重量标准化

为了搞清楚为什么需要做重量正常化，这首先从数据不均衡的问题说起，这一问题在常见的人脸识别数据集中很普遍。下图展示了在几个常用的人脸识别数据集上类别与样本数量的统计分布。



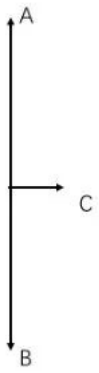
图片来自[UnequalTraining]

在[SphereFace]的附录部分，给出了关于训练数据不均衡如何影响weight的norm的一个经验分析。左边两张图是MNIST上的实验结果，右边两张图是WebFace上的实验结果。可见，对于样本数多的类别，它的重量的norm会更大。

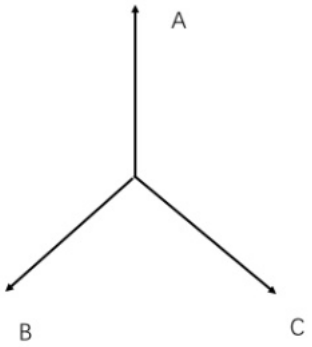


图片来自[SphereFace]

再举一个非常极端的例子。假设有一个三个人的分类问题，其中A，B两个人各有100张照片，而另外一个人C的照片只有5张，那么训练这个三分类问题，最终将其weight的每一列在二维空间上做一个可视化，就应该是下图这个样子。



对于数据量大的A，B两个人，他们的体重norm很大，几乎瓜分了整个特征空间，而C由于照片数量少，他的体重norm很小。而我们当然知道，实际情况下， A，B，C三个人绝对应该是处于一个平等的地位的，换言之，在特征空间里面，他们也应该处于一个“三足鼎立”的模式，就像下图所示。



如何让A，B，C这三个人在训练时受到平等对待呢？这个时候，Weight Normalization就起作用了。重量正常化的想法被2016年NIPS的[WeightNorm]提出。这篇论文通过大量的实验表明体重归一化相比batchnormalization能减少计算量，并且能使得网络更快的收敛。

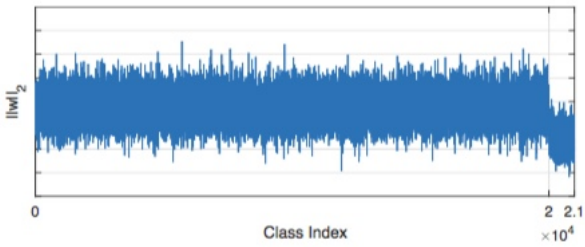
不过这篇论文并没有提到权重归一化可以用来改善数据不均衡的问题。最早把权重归一化和数据不均衡问题联系起来的是郭东尧等人的工作。郭东尧等人在2017年提出了一种做正常化的方法变种[UPLoss]，他们首先计算所有weight模长的均值，并幅值给 α

$$\alpha \leftarrow \frac{1}{|C_b|}\sum_{k\in C_b}\|\mathbf{w}_k\|_2^2$$

然后再用下面的loss约束每一个weight vector靠近这个均值

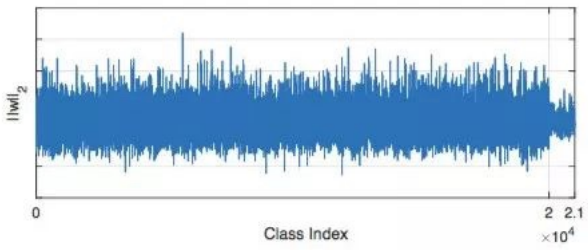
$$\mathcal{L}_{up} = \left\| \frac{1}{|C_l|}\sum_{k\in C_b}\|\mathbf{w}_k\|_2^2 - \alpha \right\|_2^2$$

作者自己建立了一个人脸数据集，包含两部分，第一部分大概包含20k个个人，每个个人有50-100张图片，第二部分包含1k个id，每个个人身份只有20张图片。如果在这个数据集上使用不加重标准化的softmax，得到的结果如下。



图片来自[UPLoss]

可以发现，上图最右侧，从2到2.1这个区间，也就是最后的1k个id的weight的norm明显小于前20k个。并且，最后1k个id在训练集上的召回只有30%。而在增加了作者提供的重量标准化方法后，最后1k个id的重量norm和前20k个id的重量norm之间的差距明显变小了。作者提到在训练集上的召唤一下子提升到了70% 。



图片来自[UPLoss]

作者在做重量正常化后在LFW上报得到的结果是99.71%，已经远远超过了不做重量normalizaton的基线98.88%。

所以说来说去，重量正常化本质上就做了一件事，在网络中引入一个先验，即告诉网络，无论类别本身的样本数量是多还是少，所有类别的地位都应该是平等的，因此它们的重量的norm也是相似的。

特征规范化

为了搞清楚为什么需要做特征规范化，我们首先看看Softmax Loss在优化的过程中会发生什么。

优化softmax loss需要增大
p_i
，也需要增大
f_i
。

为了变大

，根据上面的定义，优化会倾向于：

1. 增大正确类别的重量范数

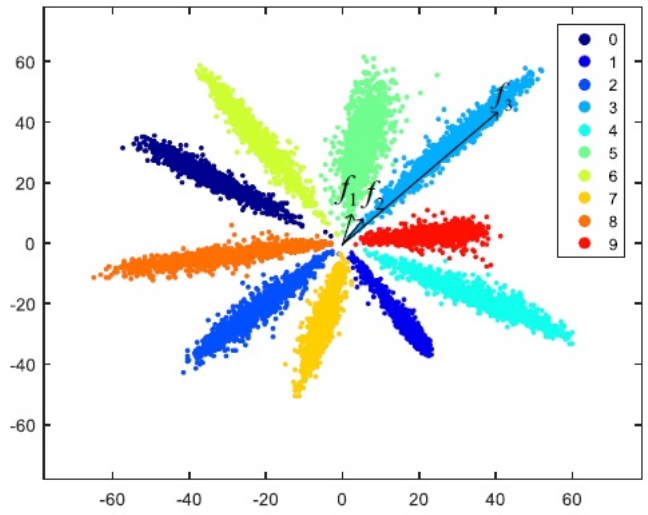
$||\mathbf{w}_{y_i}||$
。效果是样本多，简单的类别，weight norm大。反之则小。证据见[A-Softmax]的附录。

2. 增大样本的特征范数

$||\mathbf{x}_i||$
。效果是简单样本norm大，困难样本norm小。证据见很多纸，例如本文下面[L2-Softmax]的图。

3. 缩小特征和重量矢量的夹角。

数学上，上面的三种变化会同时发生，导致最终的特征分布呈现出“扇形”形式。例如，[NormFace] 中给了这样一个例子，MNIST上，特征的维度限定为2,10个类别的功能分布可视化如下：



图片来自[NormFace]

尽管分类的精度可以很高，但是这样的特征分布，对于困难样本是敏感的，推广性不好。上面的例子中，

f_2

（困难样本，norm小）和

f_3

属于同一类，但由于norm相差太大，反而和norm差不多的

f_1

（另外一类的困难样本）距离更近，于是分错了。这个例子也说明特征norm小的样本更不稳定。

同时也说明了，使用欧氏距离作为特征和特征之间的度量是不稳定的，因为它依赖于特征的规范。而如果我们用角度作为特征和特征之间差异的度量，就可以不受规范不稳定的影响。因此，我们希望网络更多的从第3点“缩小特征和重量矢量的夹角”这个方向去学习。

为了让网络从“缩小特征和重量矢量的夹角”这个方向去学习，我们自然就需要把另外两条路给堵死。最理想的情况也就是重量规范化和特征规范化都做.2017年，[CrystalLoss]（也就是[L2-Softmax]）提出了特征归一化，并做了大量的分析和实验证明这个简单的技巧对分类问题极其有效。

在 [CrystalLoss] 中，作者在MNIST上和一个简单的3个人的分类问题进行了可视化实验。在MNIST上，每个类变得更“窄”，在人脸实验上，每个类的特征变得更加集中。这些实验都证明了，使用特征规范化确实能让不同的类学出的嵌入，在角度方向上更具有可区分性。

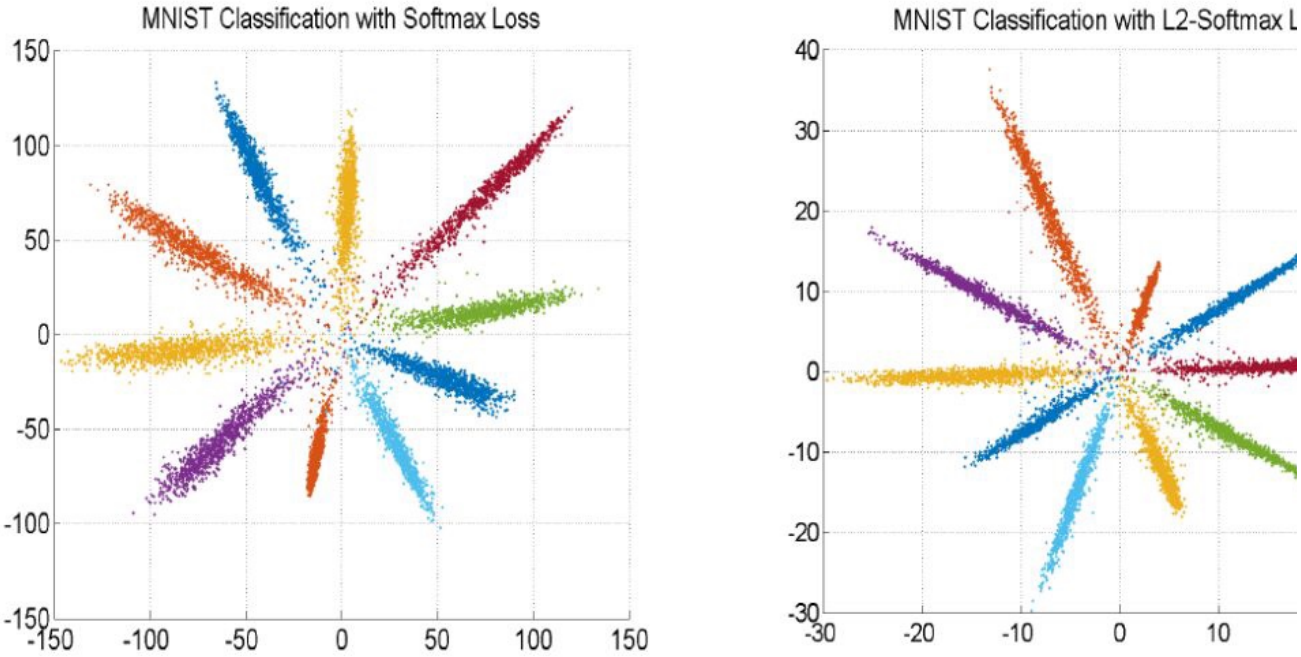


Figure 3. Vizualization of 2-dimensional features for MNIS
classification test set using (a) Softmax Loss. (b) L2-Softma

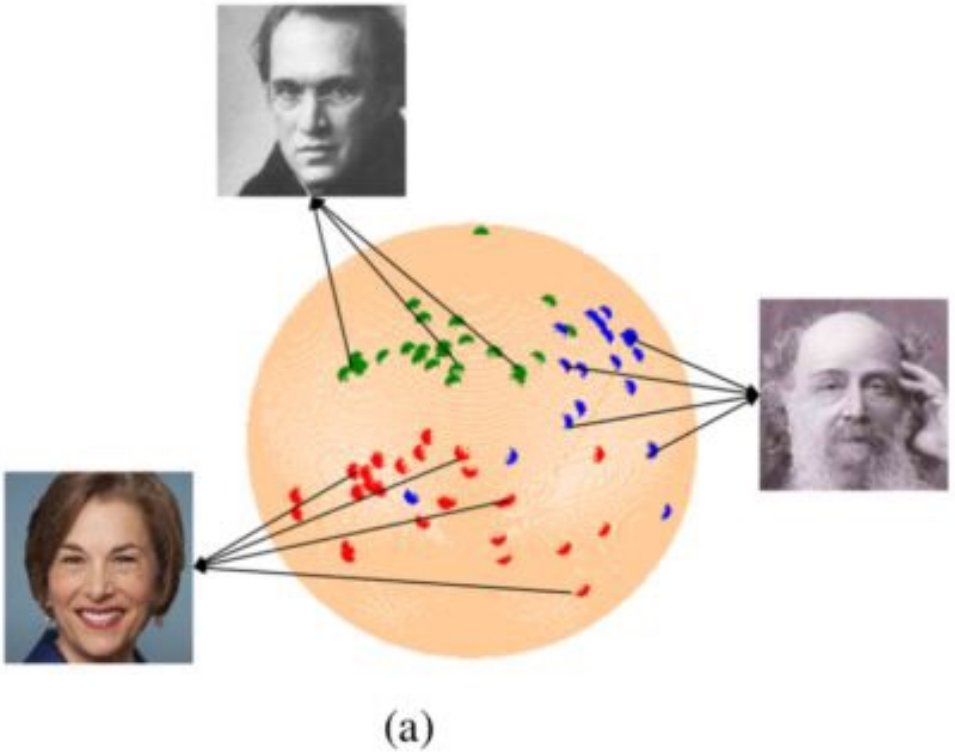


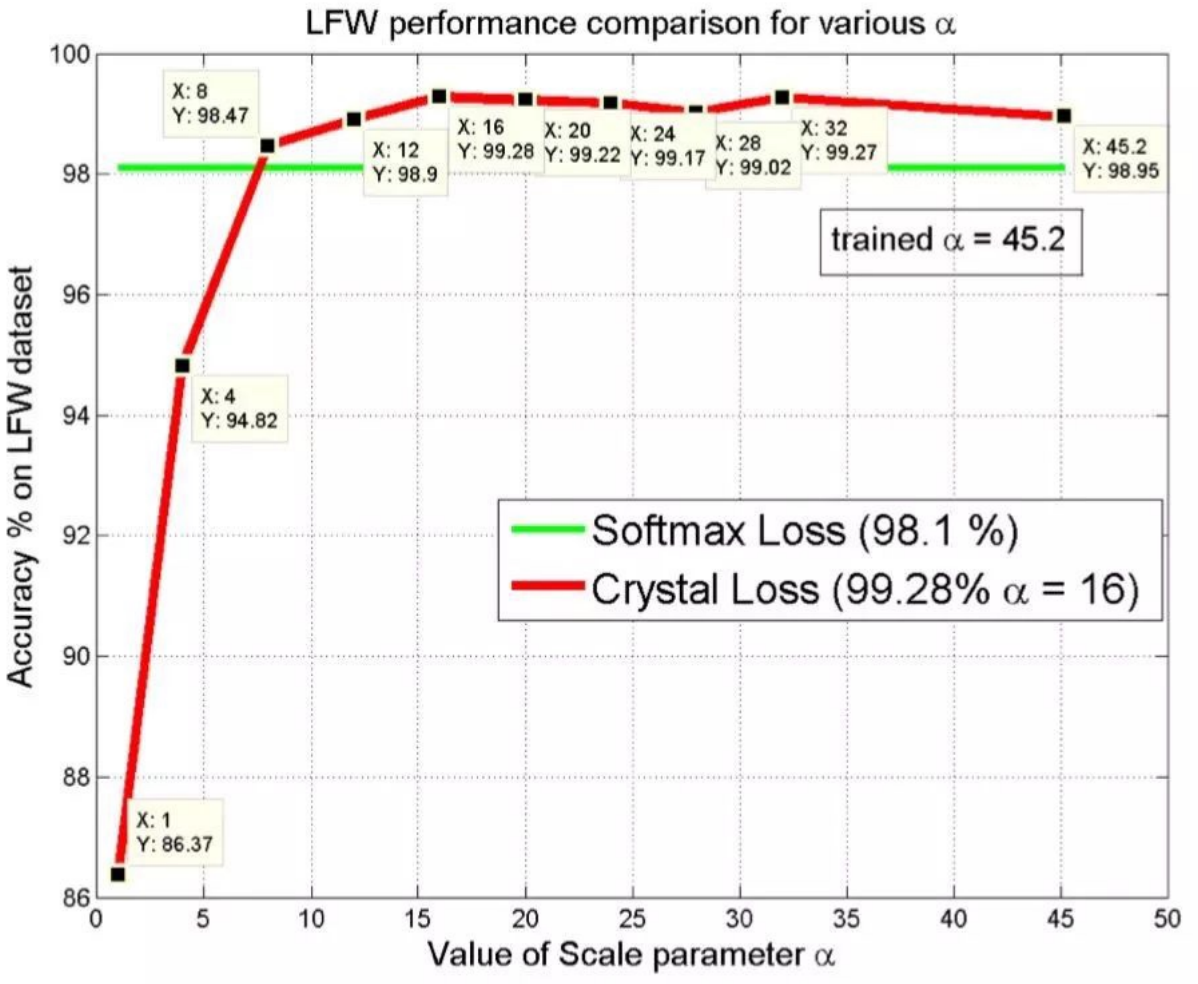
Fig. 4. Three-dimensional normalized features for three different identities, c
with Crystal Loss. The intra-class cosine distance reduces while the inter-cl

图片来自[CrystalLoss]

不过，如果只是单纯的对特征做归一化，那么极有可能陷入一个网络难以收敛的窘境，这个现象在 [L2-Softmax] 和 [NormFace] 中都提到了，并且 [NormFace] 还从数值上给出了解这个现象的解释，这里不再阐述这些细节。不过解

决方法也很简单，只需要在对功能做归化化后，再乘上一个大于1的伸缩系数，绝大多数情况下都能获得不错的收敛性。

在[L2-Softmax]　　中，对如何如何选取这个伸缩系数进行了分析和实验。一个好消息是，[L2-Softmax]　　的实验结果确实表明，网络对这个系数的选取还是非常鲁棒的。下图展示了，伸缩系数α的选取，从10左右，到50左右，网络都有不错的表现。



图片来自[CrystalLoss]

不过遗憾的是[L2-Softmax]并没有把两种normalization都加上，而只是单纯加了特征归一化。

在[L2-Softmax]的基础上，[NormFace]进一步把两种规范化都做了，并做了更详尽的分析，对规范化中的三个问题进行了全面系统的回答：

1. 为何normalization在verification中管用;
2. 为何在训练中简单做normalization不工作以及如何才能
3. 其他的度量损失如何做规范化。

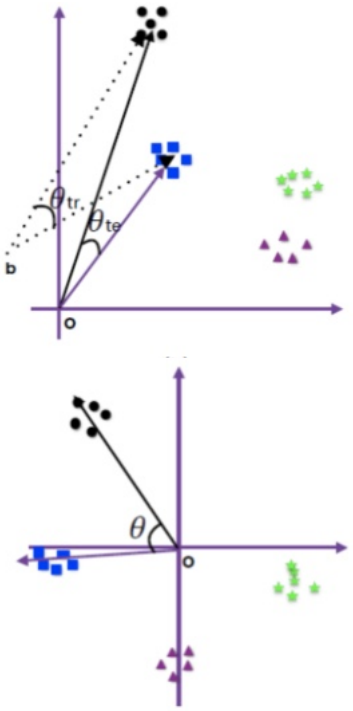
[NormFace]本身并没有提出新的想法，只是全面系统的总结了前人关于normalization的经验和技巧，并给出了令人信息的分析。

除了这些主流的两种做normalization的方法，还有一些它们的变体。在2017年的[DeepVisage]　中，提出了一种对特征做白化，而不是单纯的normalization的方法。在2018年的[CCL]中对这个想法进行了更加细致的分析和实验。对特征做白化的定义如下：

$$f^N = \frac{f^p - \mu}{\sqrt{\sigma^2}}$$

与单纯的特征归一化相比，对特征做白化的效果是希望使得所有的特征尽可能均匀分布在空间中，同时每一维发挥的作用尽量相似，不要浪费空间和维度。

作者做了一个简单的实验来展示这种归一化方式的效果，左边表示的是原始的Softmax训练后所得样本的特征在二维空间可视化后的分布，右边是使用白化后的效果。很明显能看出白化后的特征能更好地利用整个特征空间。

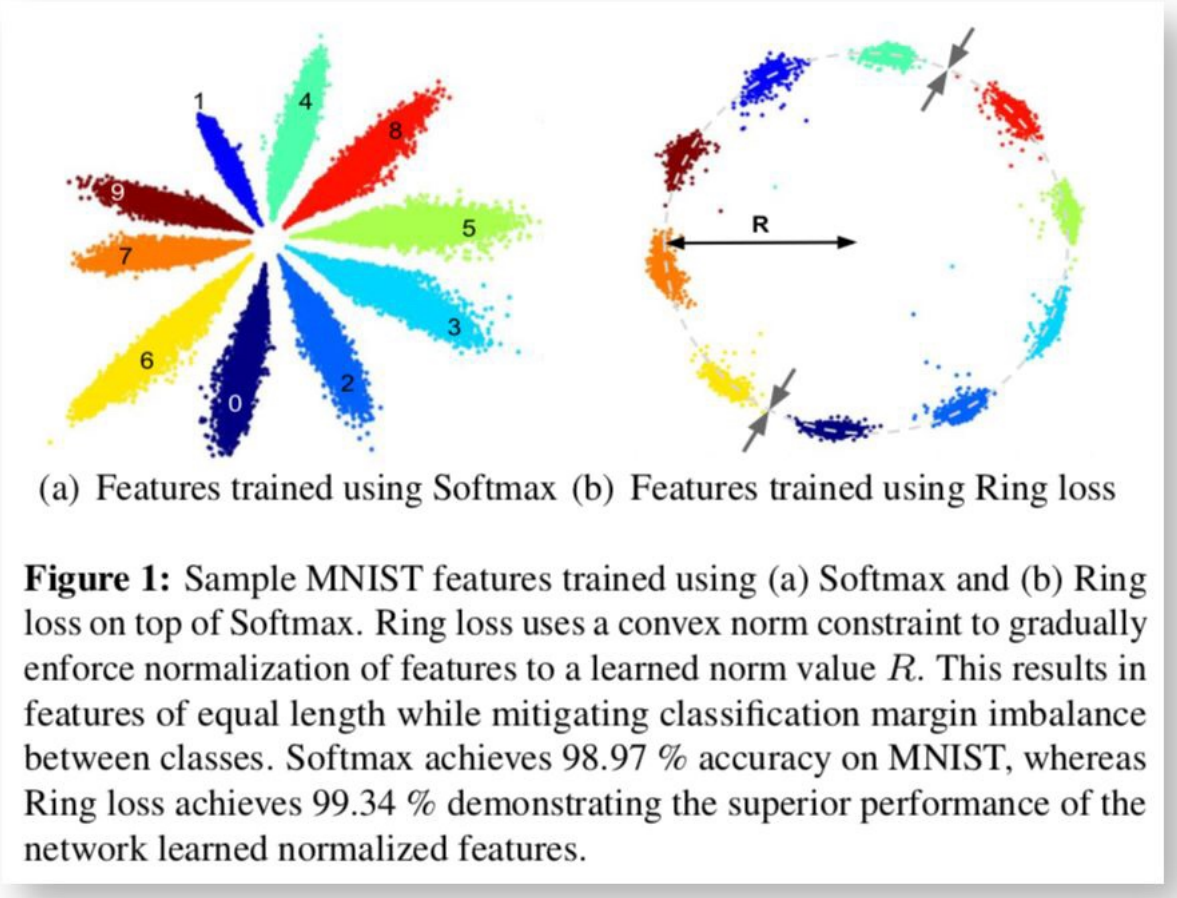


图片来自[CCL]

不过在实验部分，文章还是承认了，白化这种归一化方式，在实际使用时，由于均值很大，而方差又极小，在数值上基本上就和特征标准化很接近。

2018年CVPR的另一篇讨论特征规范化的工作是[RingLoss]。[RingLoss]　　的动机是传统的硬方式的归一化可能带来非凸的优化问题，提出的解很简单，直接加软normalization constraint到优化里面就可以了。

不过作者并没有很严格地给出传统硬方式归一化是属于非凸以及新的软方式的规范化是凸的证明，只是从一个数值实验上表面了Ring Loss具有优良的收敛性。



(a) Features trained using Softmax (b) Features trained using Ring loss

Figure 1: Sample MNIST features trained using (a) Softmax and (b) Ring loss on top of Softmax. Ring loss uses a convex norm constraint to gradually enforce normalization of features to a learned norm value R. This results in features of equal length while mitigating classification margin imbalance between classes. Softmax achieves 98.97 % accuracy on MNIST, whereas Ring loss achieves 99.34 % demonstrating the superior performance of the network learned normalized features.

图片来自[RingLoss]

Ring loss的具体形式很简单：

$$L_R = \frac{\lambda}{2n}\sum_{i=1}^{N}(\|\mathbf{x}_i\|_2 - R)^2$$

其中，R是需要学习的特征norm的参数（类似于[L2-Softmax] 的α和[NormFace]中的l， λ是控制Ring loss项的权重。

到现在为止，我们基本上已经明白了特征归一化或者权重归一化能在一定程度上控制网络对简单或者难样本的关注程度。具体一点就是，如果不加约束，网络总是希望让简单的样本的特征模长和重模长变大，让难的样本的特征和重量的模长变小，这个现象在[SphereFace] ，[NormFace] 以及[RingLoss] 中均有分析。

现在，我们知道应该把两种normalization都加上，让网络去学习角度方向上的差异性。不过如果两种normalization都做了，而不加别的处理，网络会非常难火车。在[NormFace]中，对这种现象进行了数值上的分析，指出在权和特征都归一化到1之后，在类别数较多时，即使是正确分类的样本，都将获得与分错的样本在数值上差不多大小的梯度，而这自然就会影响网络去关注那些更需要学习的样本。

[NormFace] 的给出的解决方法和[l2Softmax] 几乎一样，就是引入一个scale参数（对应[L2Softmax]中的α）。而scale参数又是如何具体的影响网络优化的过程呢？针对这个问题，[HeatedUpSoftmax] 做了非常完整的分析。并提出在两种规范化都做时，在训练的不同阶段，通过人为的控制这个规模系数α，能达到控制网络关注简单样本还是难样本的目的。

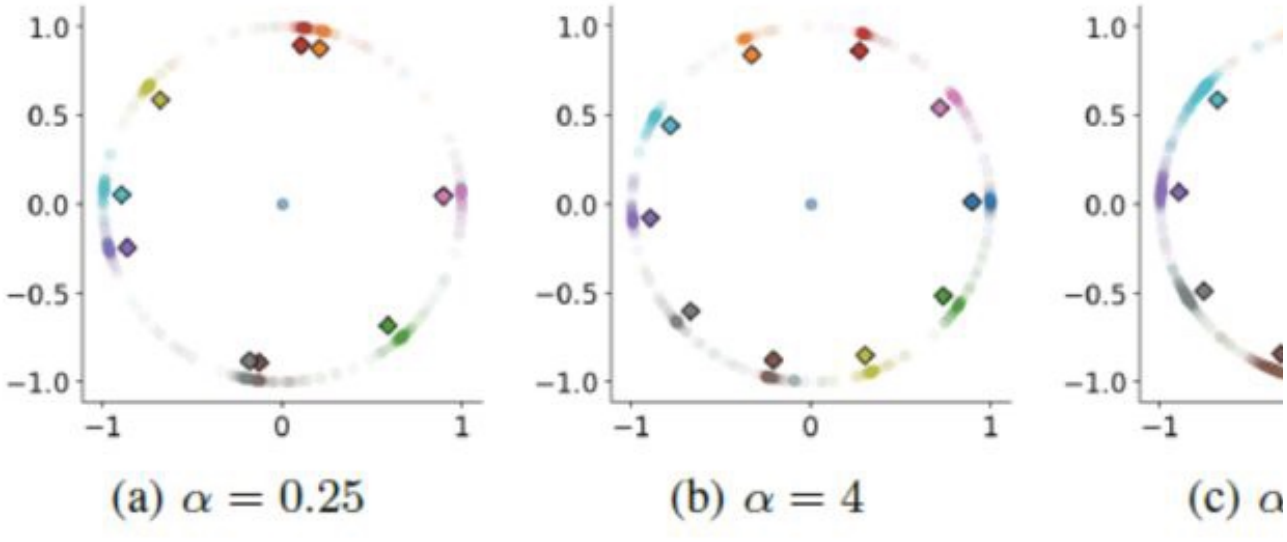
[HeatedUpSoftmax]中的Softmax形式如下。

$$p^k = \frac{\exp(\alpha f^k)}{\sum_{k'=1}^{K}\exp(\alpha f^{k'})}$$

文章给出的结论是，α从0变大的过程中

1. hard样本的梯度越来越大。
2. easy样本的梯度先小，后大，再小。

选择不同的α会影响这些样本在优化过程中的运动速度和分布情况。
下面是一个MNIST上的很好的玩具示例，显示选择不同α时的影响。



图片来自[HeatedUpSoftmax]

因此，文章提出的一种称为“HeatedUp”的训练策略，就是希望在开始的阶段，使用大的α让网络关注简单样本，从而迅速收敛，在训练的后期使用小的α让网络开始重点关注难样本。

- 情况（a）：α太小是不好的。此时所有样本都走的很慢，分类都分不开;
- 情况（d）：α太大也是不好的。此时处于分类外面很远的最难样本走的很快，能够分对。但是位于分类面附近的模棱两可的边界样本（也就是图1中的三角样本）走的不快。最终的分类面不够紧凑;
- 情况（b，c）：合适的α能让各类样本（主要是硬和边界）都获得合适的梯度，得到比较紧凑的分类面。

据此，本文提出动态调整α的策略：刚开始优化时使用大的α，让硬样本迅速变成边界样本。然后逐渐变小α，让边界样本也能获得足够的梯度，进一步缩小分类面。

逐渐变小α的过程，也就是逐渐把分类面附近的样本往类中心挤压的过程。对应地，温度逐渐增大，也就是本文的“加热”。

最后再总结一下，无论是早期的重量归一化，特征归一化，还是后期这些形式的一些变体以及一些技巧，它们归根结底，都是为了以下三点：

1. 防止网络在长尾问题上“顾此失彼”。
2. 防止网络一旦把样本分对就“浅尝辄止”。
3. 防止网络在难样本问题上“掩耳盗铃”。

增加类间角度

在真实任务中，例如面部验证，我们需要计算未知类别的样本的相似度，此时仅仅保证“已知类别分类正确”是不够的。为了更好的泛化性能，我们还需要诸如“类内样本差异小“和”类间样本差异大“这样的良好性质，而这些并不是softmax loss的直接优化目标。（尽管，softmax loss优化的好可以间接的达成这些目标。）

换而言之，除了好的分类概率，一个好的度量空间更加重要。前者成立并不意味着后者成立。如上面图的例子，在该度量空间内，欧几里德距离的性质是很差的。

Metric learning的方法会显式地优化“**类内样本差异小**”和“**类间样本差异大**”的目标，也被广泛应用于人脸识别，例如，[DeepID2]同时使用了softmax loss和对比损失（成对失败），着名的[FaceNet]仅仅使用三重态损失就能得到表现良好的特征。然而，简单度量学习是不够的。

给定N个样本，softmax loss遍历所有样本的复杂度仅为O（N），而对比损失和三重损失的复杂度为

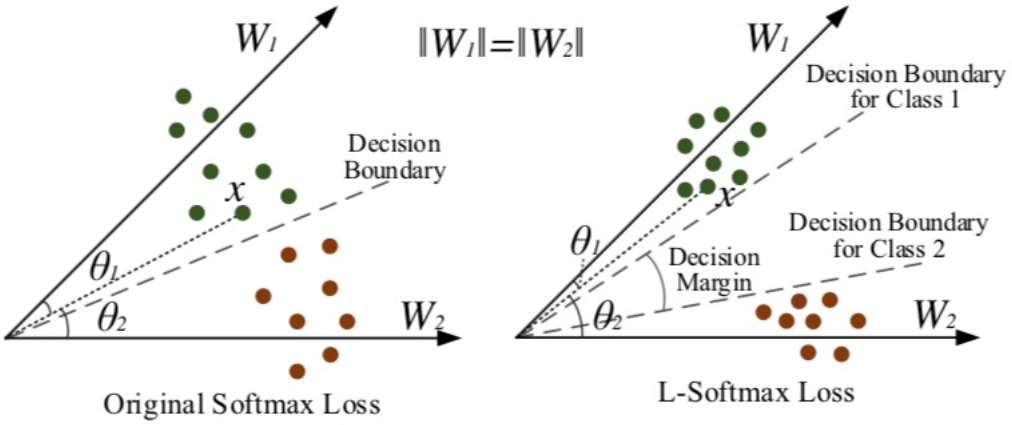
$O(N^2)$

和

$O(N^3)$

，无法简单遍历，需要有效的搜索好的训练样本，即“硬例”挖掘“问题，训练过程复杂，尤其当类别数量很大时，如何找到好的样本本身已经足够困难。（当然，此时的超大softmax loss也是个问题）。

“增加类间缘”是经典思想，但是用到softmax based loss里面是一个很有意义的创新.2016年ICML的一篇论文[L-Softmax]首次在softmax上引入了margin的概念，具有非常重大的意义。对于增加缘的形象解释，文章给出了一个很好的示意图。



图片来自[L-使用SoftMax]

$\theta_{1(2)}$

表示特征x和类权重

$\mathbf{w}_{1(2)}$

的夹角。先简化问题，假设类重量被归一化了，此时夹角就决定了样本x被分到哪一类。左边是原始的softmax loss，分界面的矢处处两

$\theta_1 = \theta_2$

类别的中间，此时 。（训练）样本会紧贴着分界面。测试的时候，就容易混淆了。右边是L-Softmax，为了在两类中间留下空白（margin），要求分界面是

$m\theta_1 = \theta_2,\ m > 1$

。

此时为了分类正确，样本特征会被压缩到一个更小的空间，两个类别的分类面也会被拉开。容易看出，此时两个类之间的角度决定边缘是

$\frac{m-1}{m+1}\theta_{1,2}$

，其中

$\theta_{1,2}$

是类重量

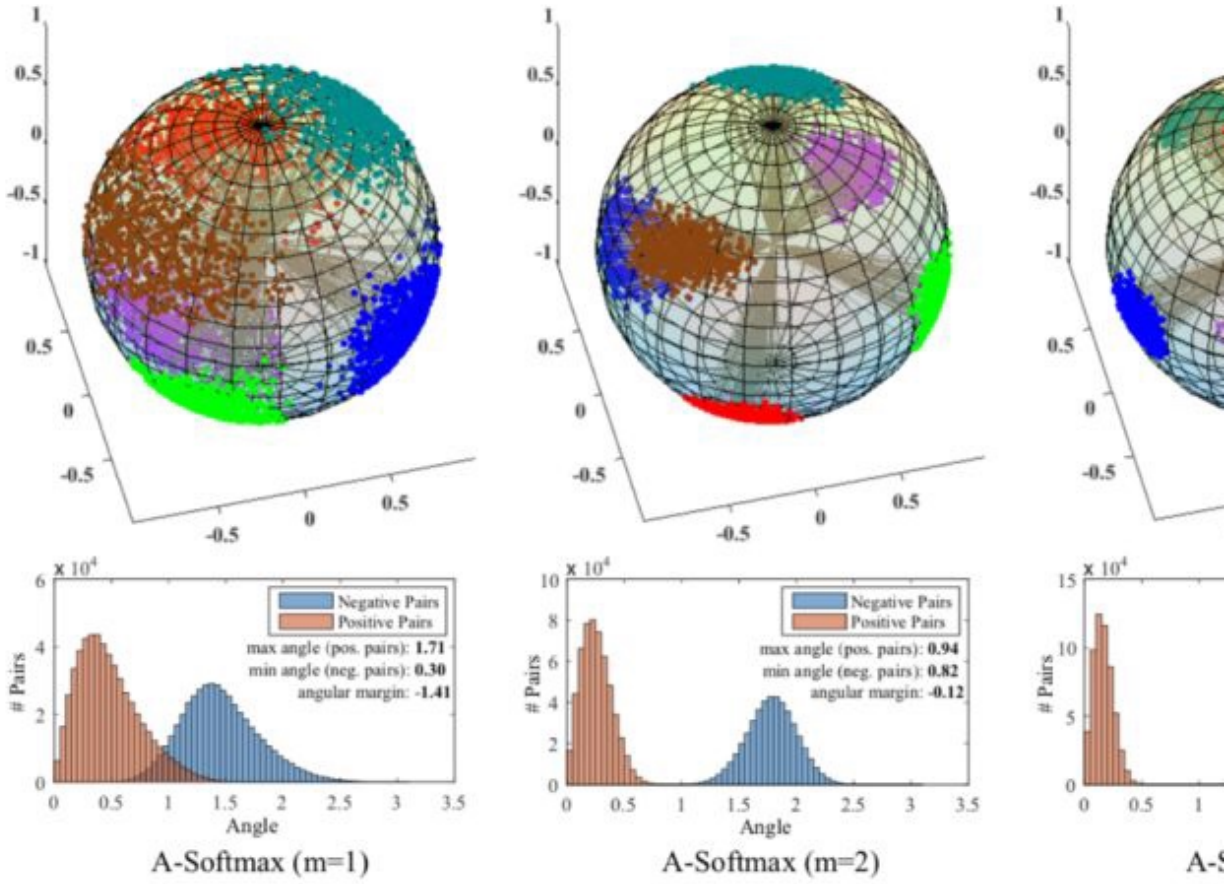
$\mathbf{w}_1$

状语从句：

$\mathbf{w}_2$

的夹角。不过可惜的是，这篇论文发表的比较早，在同时期[WeightNorm]才被发表出来，因此，在[L-使用SoftMax]中并没有引入重量Normalization.2017年CVPR的[SphereFace]在L-Softmax的基础上引入了重量标准化。

[SphereFace] 作者通过一个很形象的特征分布图，展示了引入margin的效果，可见，随着margin的增加，类内被压缩的更紧凑，类间的界限也变得更加清晰了。



图片来自[SphereFace]

[SphereFace]提出的丢失的具体形式是

$$\frac{1}{N}\sum_i -log(\frac{e^{||\mathbf{x}_i||\cdot\cos(m\cdot\theta_{y_i,i})}}{e^{||\mathbf{x}_i||\cdot\cos(m\cdot\theta_{y_i,i})}+\sum_{j\neq y_i} e^{||\mathbf{x}_i||\cdot\cos\theta_{j,i}}})$$

，这个与L2-Softmax的差别仅仅就是把当前样本与所属类别的夹角

$\theta_{y_i,i}$

变成了

$m\theta_{y_i,i}$

。

但引入边缘之后，有一个很大的问题，网络的训练变得非常非常困难。在[SphereFace]中提到需要组合退火策略等极其繁琐的训练技巧。这导致这种加缘的方式极其不实用。而事实上，这一切的困难，都是因为引入的margin是乘性缘造成的。我们来分析一下，乘性缘到底带来的麻烦是什么：

1. 第一点，乘性缘把cos函数的单调区间压小了，导致优化困难。对

$\cos(\theta_{y_i,i})$，在

处在区间[0，π]时，是一个单调函数，也就是说

落在这个区间里面的任何一个位置，网络都会朝着把

减小的方向优化。但加上乘性margin m后

的单调区间被压缩到了

$\left[0, \frac{\pi}{m}\right]$，那如果恰巧有一个样本的

落在了这个单调区间外，那网络就很难优化了；

2. 第二点，乘性缘所造成的边缘实际上是不均匀的，依赖于

的夹角。前面我们已经分析了，两个类之间的角度决定边缘

$\frac{m-1}{m+1}\theta_{1,2}$，其中

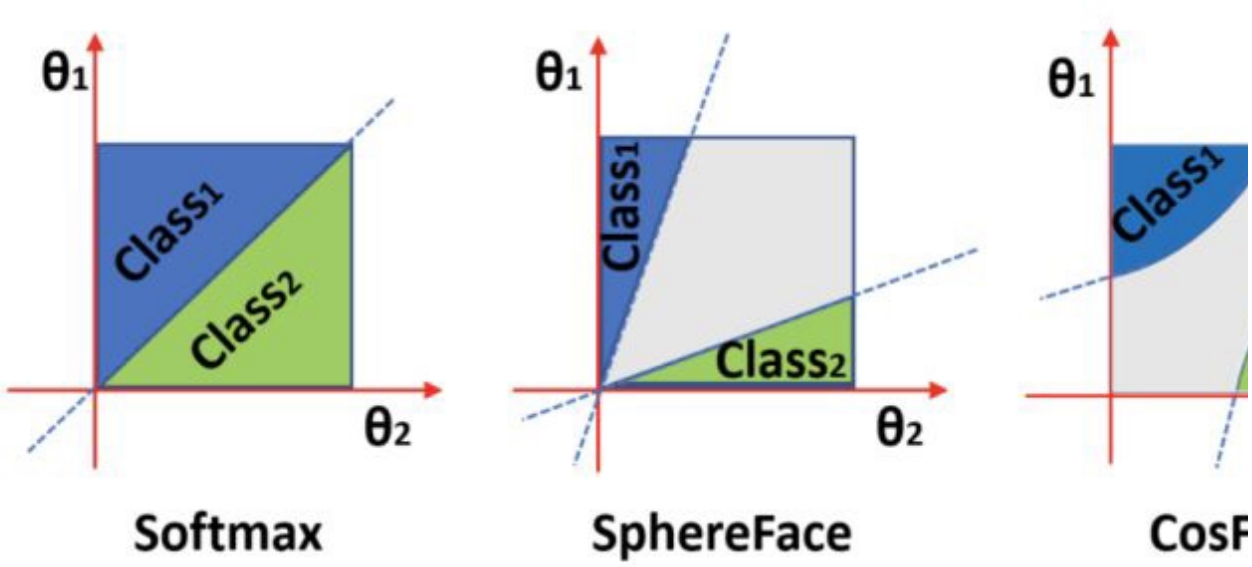
是两个类的重量的夹角。这自然带来一个问题，如果这两个类本身挨得很近，那么他们的边缘就小。特别是两个难以区分的类，可能它们的重量挨得特别近，也就是

几乎接近0，那么按照乘性边缘的方式，计算出的两个类别的间隔也是接近0的。换言之，乘性边缘易于于混淆的类不具有可分性。

为了解决这些问题，2018年ICLR的[AM-Softmax]首次将乘性的margin改成了加性的margin。虽然改动很小，但意义重大。换成加性margin之后，上述提到的两个乘性缘的弊端自然就消失了。

同时期的另一篇论文[CosFace]与之完全一样.2018年实际上还有一个工作叫[ArcFace]（做人脸识别研究的同学应该人尽皆知，业界良心），但遗憾的是在18年，[ArcFace]并没有被任何顶会收录，不过他在19年还是被CVPR 2019收录了。[ArcFace]与[AM-Softmax]同样也是加性的边缘，差别只是[ArcFace]的margin加在Cos算子的里面，而[AM-Softmax]的margin在加性算子的外面。这两者对网络的优化性能几乎一致。

在[ArcFace] 中，作者对集中加缘的方式做了很形象的对比，如下图所示。可以看出，[ArcFace]提出的边缘更符合“角度”margin的概念，而[CosFace]或是[AM-Softmax]更符合余弦边缘的概念。



图片来自[Arcface]

最后，我们总结一下加缘的几种Softmax的几种形式：

损失函数	Loss形式	决策边界
Softmax	$\frac{1}{N}\sum_{i=1}^{N}-log(\frac{e^{\mathbf{w}_{y_i}\cdot\mathbf{x}_i}}{\sum_k e^{\mathbf{w}_k\cdot\mathbf{x}_i}})$	$(W_1-W_2)x$
L-Softmax	$\frac{1}{N}\sum_i-log(\frac{e^{\|\mathbf{w}_{y_i}\|\cdot\|\mathbf{x}_i\|\cdot\cos(m\cdot\theta_{y_i,i})}}{e^{\|\mathbf{w}_{y_i}\|\cdot\|\mathbf{x}_i\|\cdot\cos(m\cdot\theta_{y_i,i})}+\sum_{k\neq y_i}e^{\|\mathbf{w}_k\|\cdot\|\mathbf{x}_i\|\cdot\cos\theta_{k,i}}})$	$(\|W_1\|\cos m\theta_1-\|W_2\|\cos\theta_2)=0$
A-Softmax	$\frac{1}{N}\sum_i-log(\frac{e^{\|\mathbf{x}_i\|\cdot\cos(m\cdot\theta_{y_i,i})}}{e^{\|\mathbf{x}_i\|\cdot\cos(m\cdot\theta_{y_i,i})}+\sum_{j\neq y_i}e^{\|\mathbf{x}_i\|\cdot\cos\theta_{j,i}}})$	$\|x\|(\cos m\theta_1$
AM-Softmax	$\frac{1}{N}\sum_i-log(\frac{e^{s\cdot(\cos\theta_{y_i,i}-m)}}{e^{s\cdot(\cos\theta_{y_i,i}-m)}+\sum_{k\neq y_i}e^{s\cdot\cos\theta_{k,i}}})$	$s(\cos\theta_1-m$
ArcFace	$\frac{1}{N}\sum_i-log(\frac{e^{s\cdot\cos(\theta_{y_i,i}+m)}}{e^{s\cdot\cos(\theta_{y_i,i}+m)}+\sum_{k\neq y_i}e^{s\cdot\cos\theta_{k,i}}})$	$s(\cos(\theta_1+m$

总结

Softmax相关的两大关键主题，做归一化以及增加margin。通过归一化的技巧，极大缓解了传统Softmax在简单与困难样本间“懒惰学习”的问题以及长尾数据造成的类间不平衡问题。通过增加margin，使得Softmax Loss能学习到更具有区分性的度量空间。但到这里问题还远远没有结束，现存的问题有：

1. 在归一化技巧下，嘈杂的样品对网络的负面干扰也被放大，如何削弱其影响值得进一步思索;
2. 即使做了重归一化，长尾问题也只是得到一定的缓解，不平衡的问题依然存在;
3. 增加边缘虽然让网络学到了更好的度量空间，但引入的超参到底怎么样才是最优的选项?

这些问题依然还没有被很好解决。

参考文献

1. [DeepID2]：[2014，NIPS]，[通过联合识别 - 验证的深度学习面部表示]
2. [FaceNet]：[2015，CVPR]，[FaceNet：用于人脸识别和聚类的统一嵌入]
3. [WeightNorm]：[2016，NIPS]，[权重归一化：加速深度神经网络训练的简单重新参数化]
4. [L-Softmax]：[2016，ICML]，[卷积神经网络的大边界Softmax损失]
5. [DeepVisage]：[2017，arxiv]，[DeepVisage：使用强大的泛化技能简化人脸识别]
6. [L2-Softmax]：[2017，arxiv]，[L2-constrained Softmax Loss for Discrimtive Face Reification]
7. [SphereFace]：[2017，CVPR]，[SphereFace：用于人脸识别的深层超球面嵌入]
8. [UPLoss]：[2017，arxiv]，[通过推广代表性不足的课程进行一次性面部识别]
9. [NormFace]：[2017，ACM MultiMedia]，[NormFace：用于面部验证的L2超球面嵌入]
10. [AM-Softmax]：[2018年，ICLR研讨会]，[面部验证的附加保证金Softmax]
11. [CCL]：[2018，arxiv]，[通过集中协调学习进行面部识别]
12. [ArcFace]：[2018，arxiv]，[ArcFace：深层人脸识别的附加角度边缘损失]
13. [CosFace]：[2018，CVPR]，[CosFace：深度人脸识别的大边际余弦损失]
14. [RingLoss]：[2018，CVPR]，[环损：面部识别的凸面特征标准化]
15. [CrystalLoss]：[2018年，期刊]，[无限制面部验证和识别的水晶损失和质量汇集]
16. [FaceRecSurvey]：[2018]，[深度面部识别：调查]
17. [HeatedUpSoftmax]：[2018]，[加热Softmax嵌入]
18. [UnequalTraining]：[2019年CVPR]，[用于长尾噪声数据的深度人脸识别的不平等训练]
19. [人脸识别的LOSS（上，下）]：[知乎]，[https://zhuanlan.zhihu.com/p/34404607,https://zhuanlan.zhihu.com/p/34436551]
20. [人脸识别最前沿在研究什么？]：[知乎]，[https://www.zhihu.com/question/67919300/answer/304674212]

-完-

*延伸阅读

- 卷积神经网络系列之softmax，softmax loss和cross entropy的讲解
- 从最优化的角度看待 Softmax 损失函数
- 一文道尽softmax loss及其变种

添加极市小助手微信（**ID：cv-mart**），备注：**研究方向-姓名-学校/公司-城市**（如：目标检测-小极-北大-深圳），即可申请加入**目标检测、目标跟踪、人脸、工业检测、医学影像、三维&SLAM、图像分割等极市技术交流群**，更有每月**大咖直播分享、真实项目需求对接、干货资讯汇总，行业技术交流**，一起来让思想之光照的更远吧~

△长按添加极市小助手

△长按关注极市平台

觉得有用麻烦给个在看啦~