# 人脸识别中Softmax-based Loss的演化史



本文授权转载自：旷视研究院

**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



参考文献


1. [*DeepID2*]：[2014, NIPS]，[Deep Learning Face Representation by Joint Identification-Verification]
2. [*FaceNet*]：[2015, CVPR]，[FaceNet: A Unified Embedding for Face Recognition and Clustering]
3. [*WeightNorm*]：[2016, NIPS]，[Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks]
4. [*L-Softmax*]：[2016, ICML]，[Large-Margin Softmax Loss for Convolutional Neural Networks]
5. [*DeepVisage*]：[2017, arxiv]，[DeepVisage: Making face recognition simple yet with powerful generalization skills]
6. [*L2-Softmax*]：[2017, arxiv]，[L2-constrained Softmax Loss for Discriminative Face Verification]
7. [*SphereFace*]：[2017, CVPR]，[SphereFace: Deep Hypersphere Embedding for Face Recognition]
8. [UPLoss]：[2017, arxiv]，[One-shot Face Recognition by Promoting Underrepresented Classes]
9. [*NormFace*]：[2017, ACM MultiMedia]，[NormFace: L2 Hypersphere Embedding for Face Verification]
10. [*AM-Softmax*]：[2018, ICLR workshop]，[Additive Margin Softmax for Face Verification]
11. [*CCL*]：[2018, arxiv]，[Face Recognition via Centralized Coordinate Learning]
12. [*ArcFace*]：[2018, arxiv]，[ArcFace: Additive Angular Margin Loss for Deep Face Recognition]
13. [*CosFace*]：[2018, CVPR]，[CosFace: Large Margin Cosine Loss for Deep Face Recognition]
14. [*RingLoss*]：[2018, CVPR]，[Ring loss: Convex Feature Normalization for Face Recognition]
15. [*CrystalLoss*]：[2018, a journal]，[Crystal Loss and Quality Pooling for Unconstrained Face Verification and Recognition]
16. [*FaceRecSurvey*]：[2018]，[Deep Face Recognition: A Survey]
17. [*HeatedUpSoftmax*]：[2018]，[Heated-Up Softmax Embedding]
18. [UnequalTraining]：[2019 CVPR]，[Unequal-Training for Deep Face Recognition With Long-Tailed Noisy Data]
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]


## 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 paper 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 paper 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。

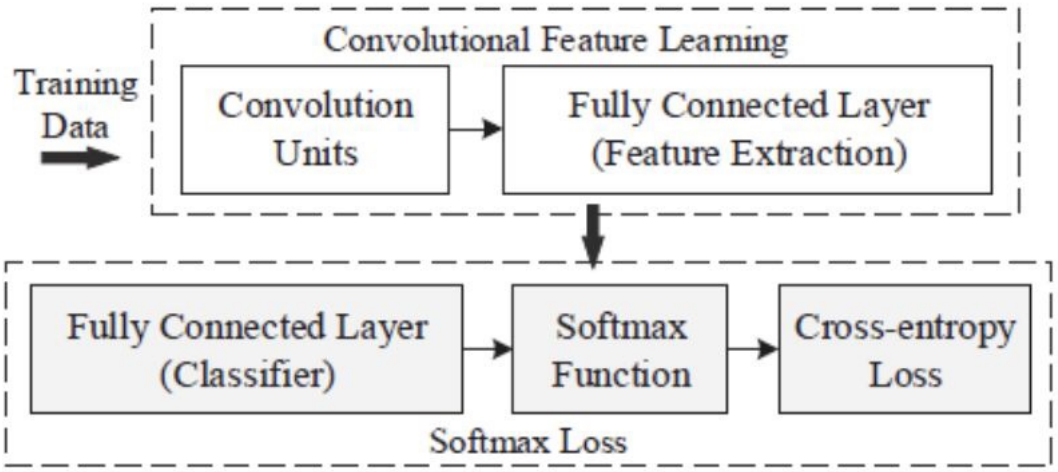
## Softmax 简介

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 feature 拿到 retrieval，verification 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 verification 任务所需要的是一个泛化性能更好的度量空间（metric space）。保证分类正确和保证一个泛化性优良的 metric space 这两者之间虽然相关性很强，但并不直接等价。

因此，近年来，face recognition 领域的主要技术进展集中在如何改进 softmax 的 loss，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 metric 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 margin。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 feature extraction layer 输出 feature x，和最后一层的 classification layer 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 function 得到 normalize 后的类别概率，再得到 cross-entropy loss。




图片来自*[FaceRecSurvey]*

类别 weight
$\mathbf{w}_k$
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它类别分数，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（Normalization）

归一化(normalization)，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 feature 做归一化（feature normalization）的定义为

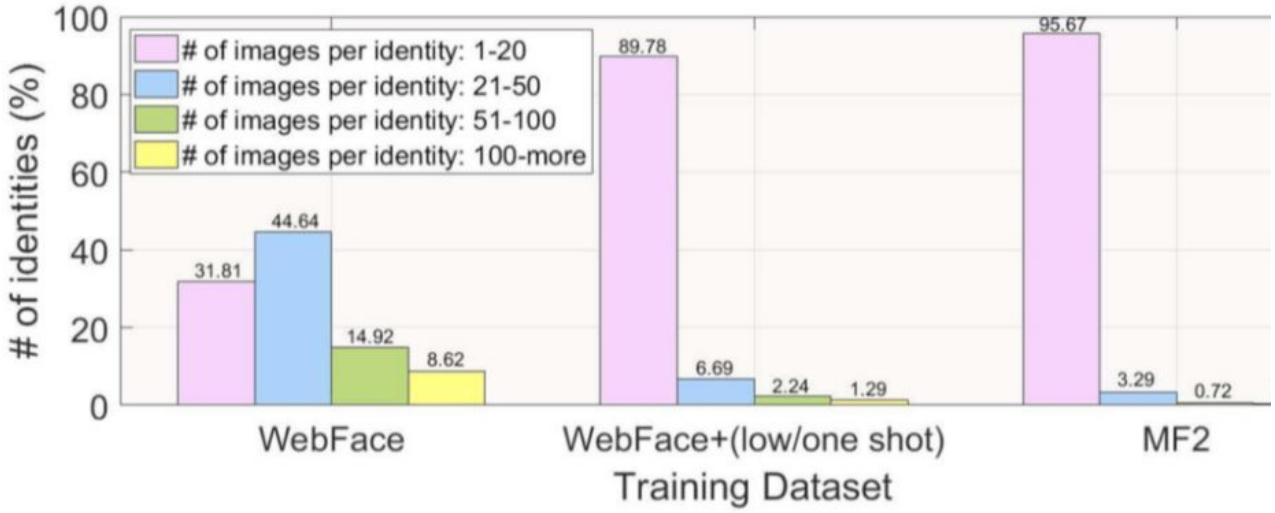
$$\mathbf{x} \rightarrow \frac{\mathbf{x}}{||\mathbf{x}||_2}$$

- 对 weight 做归一化(weight normalization）的定义为

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

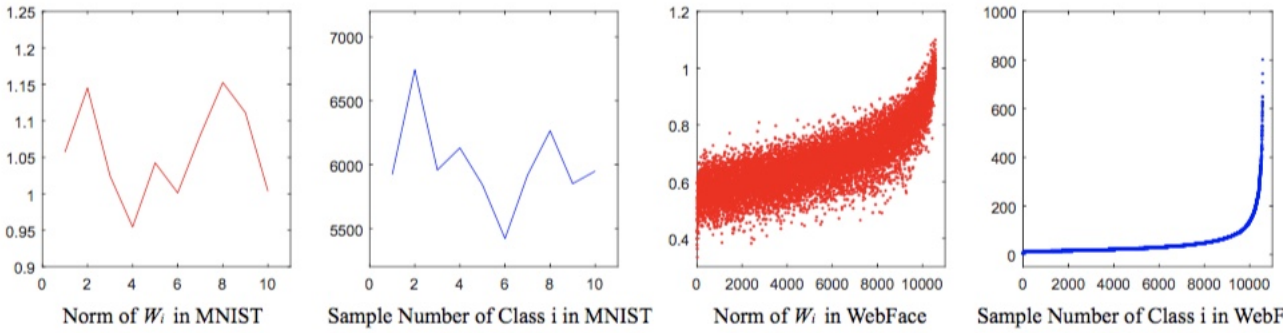
### Weight Normalization

为了搞清楚为什么需要做 weight normalization，这首先要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 sample 数量的统计分布。



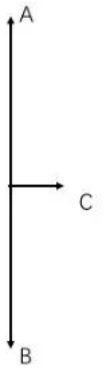

图片来自*[UnequalTraining]*

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 weight 的 norm 会更大。

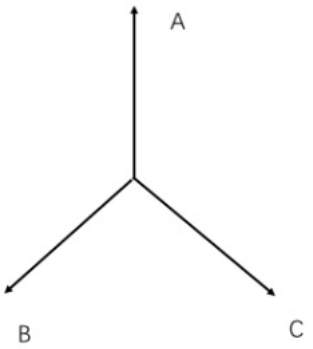



图片来自*[SphereFace]*

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 weight 的 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 weight 的 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。Weight Normalization 的想法被 2016 年 NIPS 的 *[WeightNorm]* 提出。这篇paper通过大量的实验表明了 Weight normalization 相比 batchnormalization 能减少计算量，并且能使得网络更快的收敛。

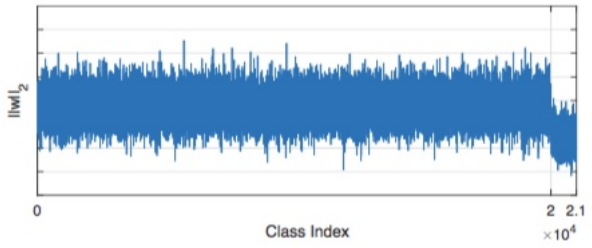
不过这篇 paper 并没有提到 weight normalization 可以用来改善数据不均衡的问题。最早把 weight normalization 和数据不均衡问题联系起来的是 Guo Dongyao 等人的工作。Guo Dongyao 等人在 2017 年提出了一种做 weight normalization 的方法变种 *[UPLoss]*，他们首先计算所有 weight 模长的均值，并幅值给 α

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

然后再用下面的 loss 约束每一个 weight vector 靠近这个均值

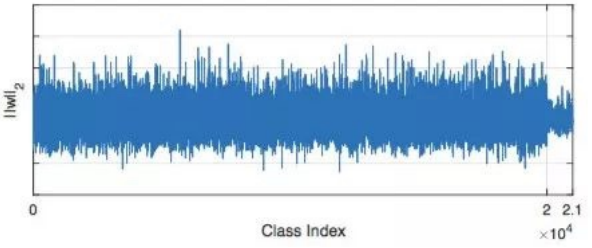
$$\mathcal{L}_{up} = \left\| \frac{1}{|Cl|} \sum_{k \in C_b} \|\mathbf{w_k}\|_2^2 - \alpha \right\|_2^2$$

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 id，每个 id 有 50-100 张图片，第二部分包含 1k 个 id，每个 id 仅有 20 张图片。如果在这个数据集上使用不加 weight normalization 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 recall 只有 30%。而在增加了作者提供的 weight normalization 方法后，最后 1k 个 id 的 weight norm 和前 20k 个 id 的 weight norm 之间的差距明显变小了。作者提到在训练集上的 recall 一下子提升到了 70%。




图片来自*[UPLoss]*

作者在做了 weight normalization 后在 LFW 上 report 得到的结果是 99.71%，已经远远超过了不做 weight normalizaton 的 baseline 98.88%。

所以说来说去， weight normalization 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 sample 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 weight 的 norm 也是相似的。

## Feature Normalization

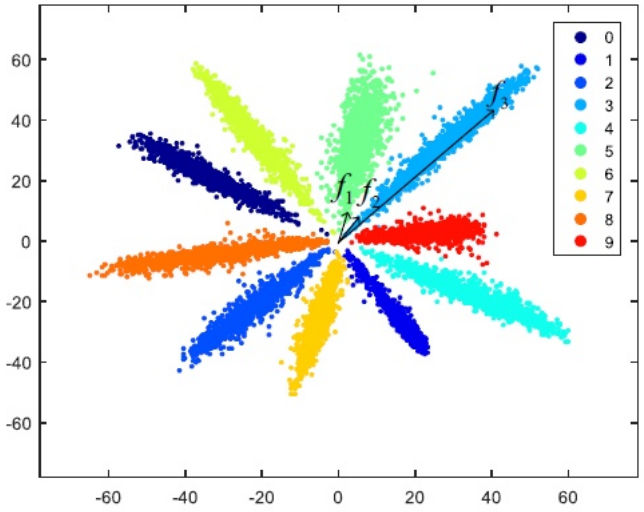
为了搞清楚为什么需要做 feature normalization，我们首先看看 Softmax Loss 在优化的过程中会发生什么。

优化 softmax loss 需要增大 $p_i$，也需要增大 $f_i$。

为了变大 $||\mathbf{w}_{y_i}||$，根据上面的定义，优化会倾向于：

1. 增大正确类别的 weight norm $||\mathbf{w}_{y_i}||$。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 feature norm $||\mathbf{x}_i||$。效果是简单样本 norm 大，困难样本 norm 小。证据见很多 paper，例如本文下面 *[L2-Softmax]* 的 Figure。
3. 缩小 feature 和 weight vector 的夹角。

数学上，上面的三种变化会同时发生，导致最终的 feature 分布呈现出“扇形”形式。例如， *[NormFace]* 中给了这样一个例子，MNIST 上，feature 的维度限定为 2，10 个类别的 feature 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 feature 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 feature norm 小的样本更不稳定。

同时也说明了，使用欧氏距离作为 feature 和 feature 之间的度量是不稳定的，因为它依赖于 feature 的 norm。而如果我们用角度作为 feature 和 feature 之间差异的度量，就可以不受 norm 不稳定的影响。因此，我们希望网络更多的从第 3 点“缩小 feature 和 weight vector 的夹角”这个方向去学习。

为了让网络从“缩小 feature 和 weight vector 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 weight normalization 和 feature normalization 都做。2017 年，*[CrystalLoss]*（也就是 *[L2-Softmax]*）提出了 feature normalization，并做了大量的分析和实验证明这个简单的 trick 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 feature 变得更加集中。这些实验都证明了，使用 feature normalization 确实能让不同的类学出的 embedding，在角度方向上更具有可区分性。

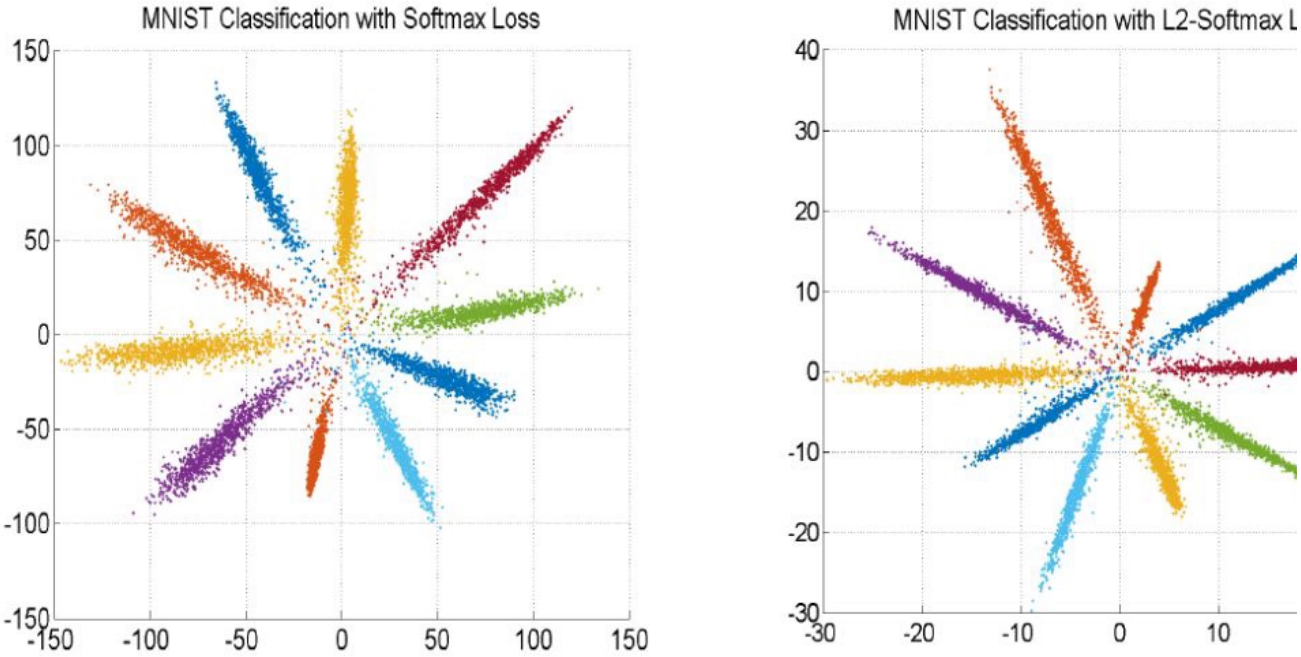



Figure 3. Vizualization of 2-dimensional features for MNIS
classification test set using (a) Softmax Loss. (b) L2-Softma

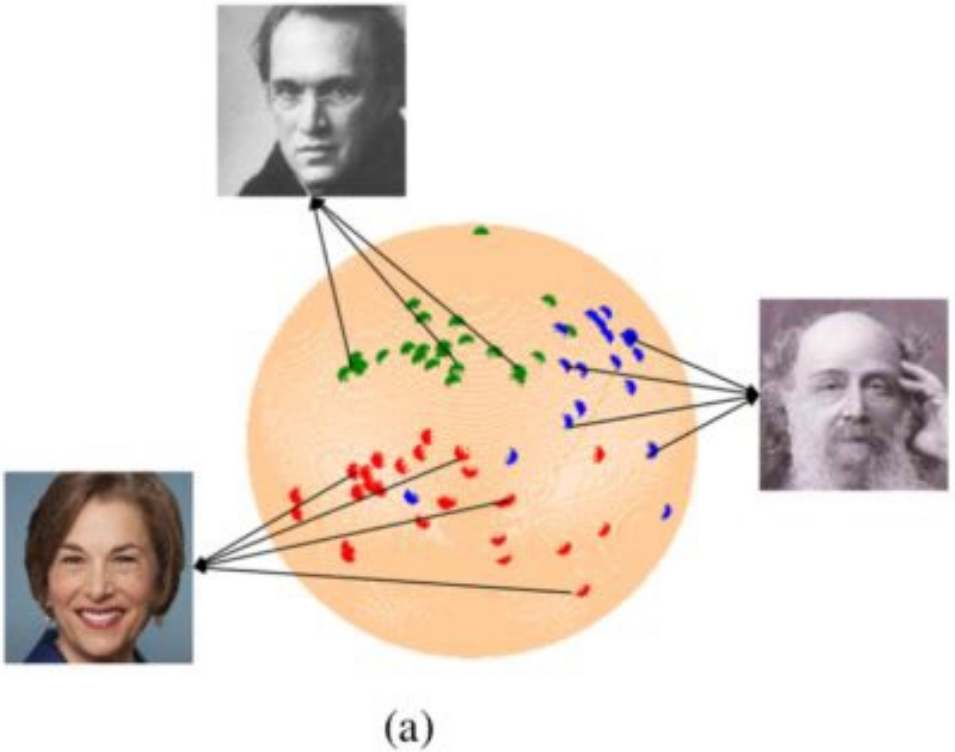


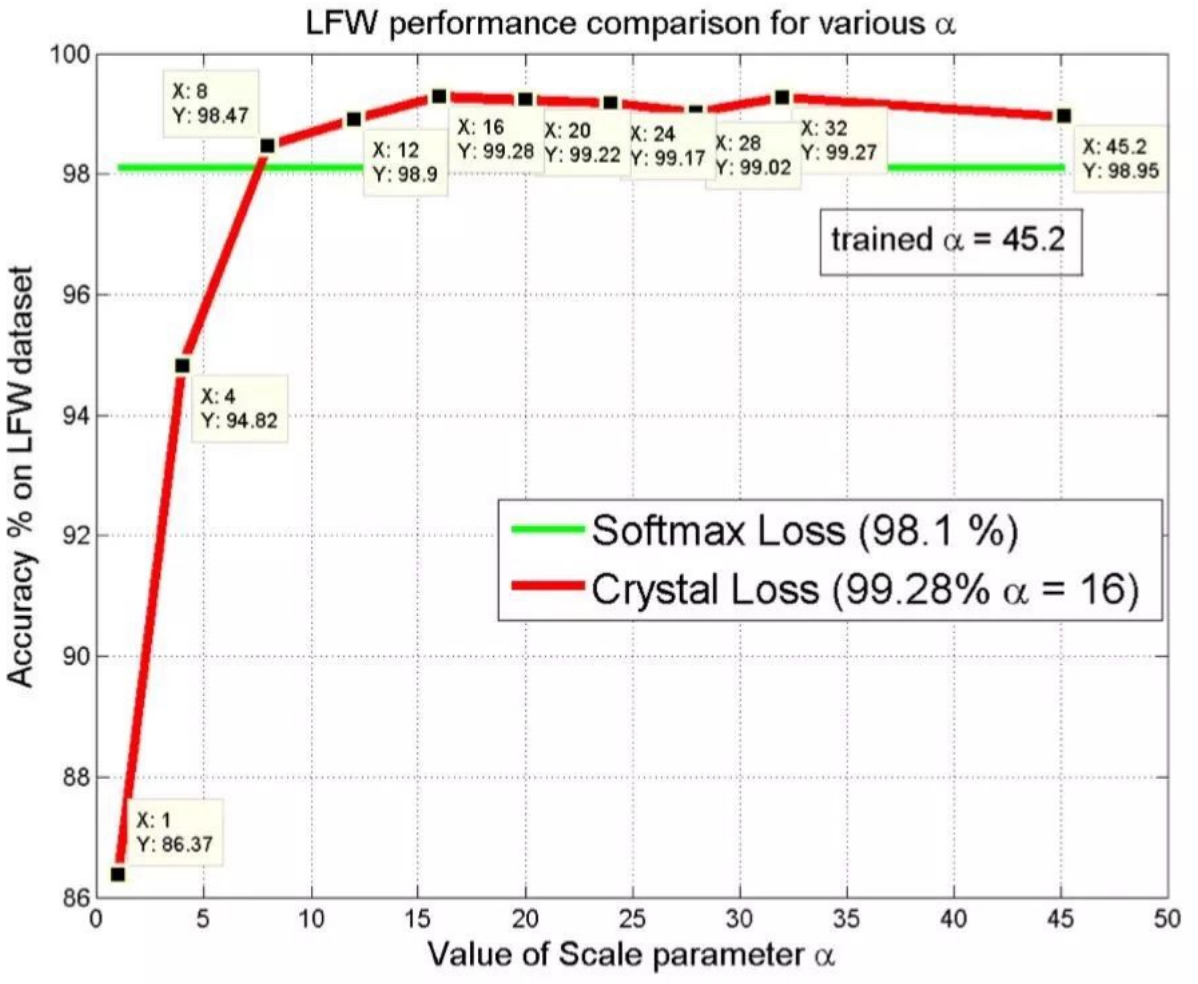

Fig. 4. Three-dimensional normalized features for three different identities, c
with Crystal Loss. The intra-class cosine distance reduces while the inter-cl

图片来自*[CrystalLoss]*

不过，如果只是单纯的对 feature 做 normalization，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 feature 做归一化后，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 feature normalization。
在 *[L2-Softmax]* 的基础上，*[NormFace]* 进一步把两种 normalization 都做了，并做了更详尽的分析，对 normalization 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 work 以及如何才能 work;
3. 其他的 metric loss 如何做 normalization。

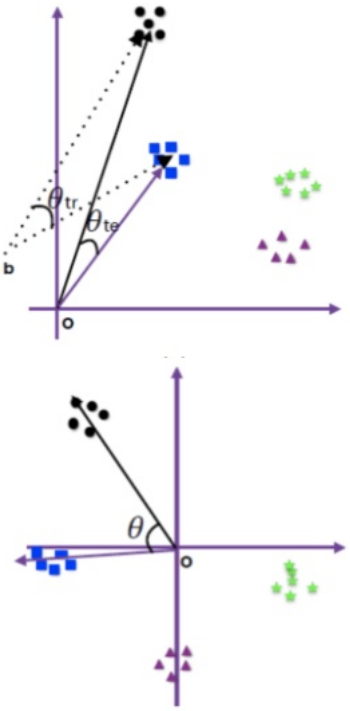
*[NormFace]* 本身并没有提出新的 idea，只是全面系统的总结了前人关于 normalization 的经验和技巧，并给出了令人信服的分析。

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 feature 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 feature 做白化的定义如下：

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

与单纯的 feature normalization 相比，对 feature 做白化的效果是希望使得所有的 feature 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 feature 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 feature 能更好地利用整个 feature 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 feature normalization 很接近。

2018 年 CVPR 的另一篇讨论 feature normalization 的工作是 *[RingLoss]*。*[RingLoss]* 的 Motivation 是传统的 hard 方式的归一化可能带来 non-convex 的优化问题，提出的 solution 很简单，直接加 soft 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 hard 方式归一化是属于 non-convex 以及新的 soft 方式的 normalization 是 convex 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。

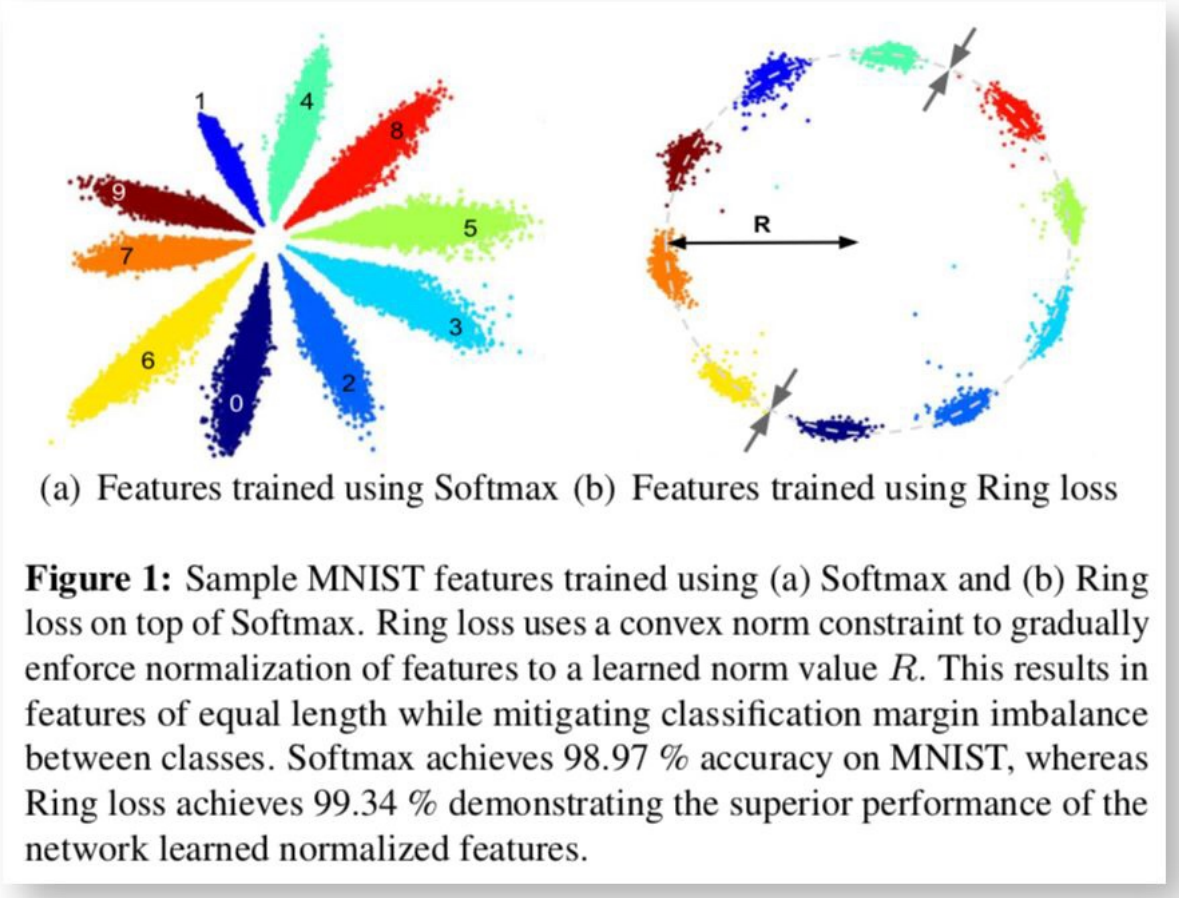



**Figure 1:** Sample MNIST features trained using (a) Softmax and (b) Ring loss on top of Softmax. Ring loss uses a convex norm constraint to gradually enforce normalization of features to a learned norm value $R$. This results in features of equal length while mitigating classification margin imbalance 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 feature 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 feature normalization 或者 weight normalization 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 feature 模长和 weight 模长变大，让难的样本的 feature 和 weight 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 train。在 *[NormFace]* 中，对这种现象进行了数值上的分析，指出在 weight 和 feature 都归一化到 1 之后，在类别数较多时，即使是正确分类的 sample，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 normalization 都做时，在训练的不同阶段，通过人为的控制这个 scale 系数 α，能达到控制网络关注简单样本还是难样本的目的。

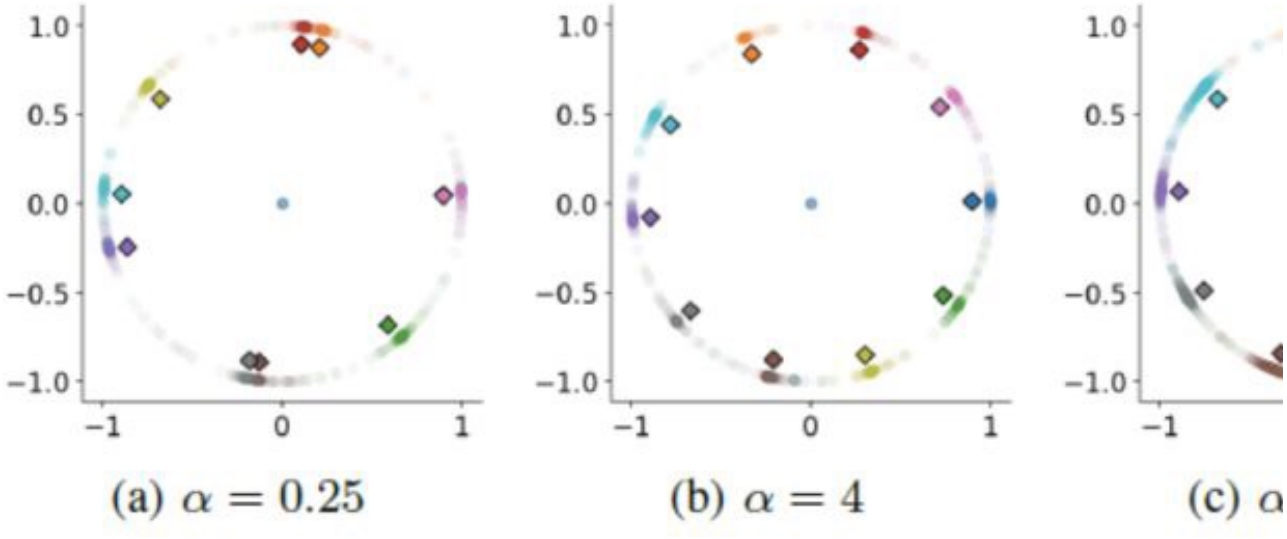
*[HeatedUpSoftmax]* 中的 Softmax 形式如下。

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

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 toy example，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hard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 boundary 样本（也就是 Figure 1 中的 triangle 样本）走的不快。最终的分类面不够 compact；
- 情况 (b, c)：合适的 α 能让各类样本（主要是 hard 和 boundary）都获得合适的梯度，得到比较 compact 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 hard 样本迅速变成 boundary 样本。然后逐渐变小 α，让 boundary 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，temperature 逐渐增大，也就是本文的“heated up”。

最后再总结一下，无论是早期的 weight normalization，feature normalization，还是后期这些形式的一些变体以及一些 trick，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 angular margin

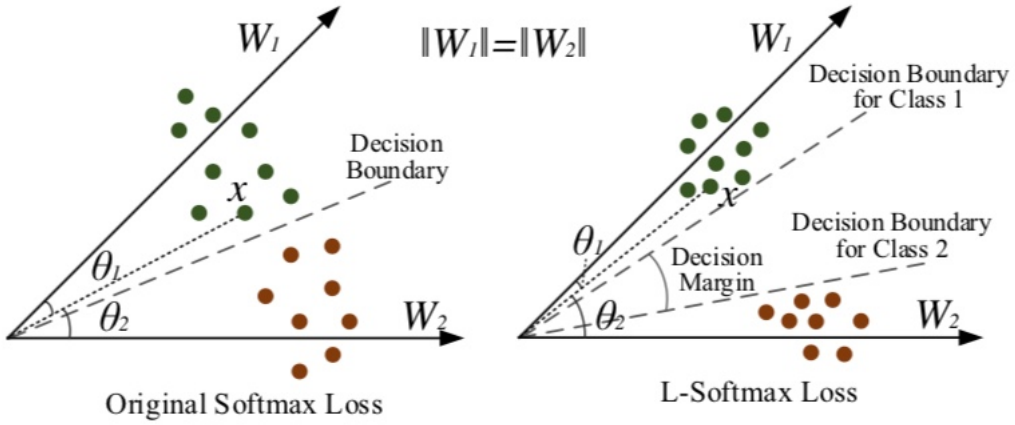
在真实任务中，例如 face verification，我们需要计算未知类别的样本的相似度，此时仅仅保证“已知类别分类正确”是不够的。为了更好的泛化性能，我们还需要诸如“类内样本差异小”和“类间样本差异大”这样的良好性质，而这些并不是 softmax loss 的直接优化目标。（尽管，softmax loss 优化的好可以间接的达成这些目标。）

换而言之，除了好的 classification probability，**一个好的 metric space 更加重要**。前者成立并不意味着后者成立。如上面 Figure 的例子，在该 metric space 内，Euclidean distance 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 face recognition，例如，*[DeepID2]* 同时使用了 softmax loss 和 contrastive loss （pairwise loss），著名的 [FaceNet] 仅仅使用 triplet loss 就能得到表现良好的 feature。然而，简单 Metric Learning 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 contrastive loss 和 triplet loss 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“hard example mining”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 margin”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 paper *[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 margin 的形象解释，文章给出了一个很好的示意图。




图片来自*[L-Softmax]*

$\theta_{1(2)}$ 表示 feature　x 和 class weight $\mathbf{w}_{1(2)}$ 的夹角。先简化问题，假设 class weight 被 normalize 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 vector 处于两类 weight 的中间，此时 $\theta_1 = \theta_2$。（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 feature 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 class 之间的 angular decision margin 是 $\frac{m-1}{m+1}\theta_{1,2}$，其中 $\theta_{1,2}$

是 class weight

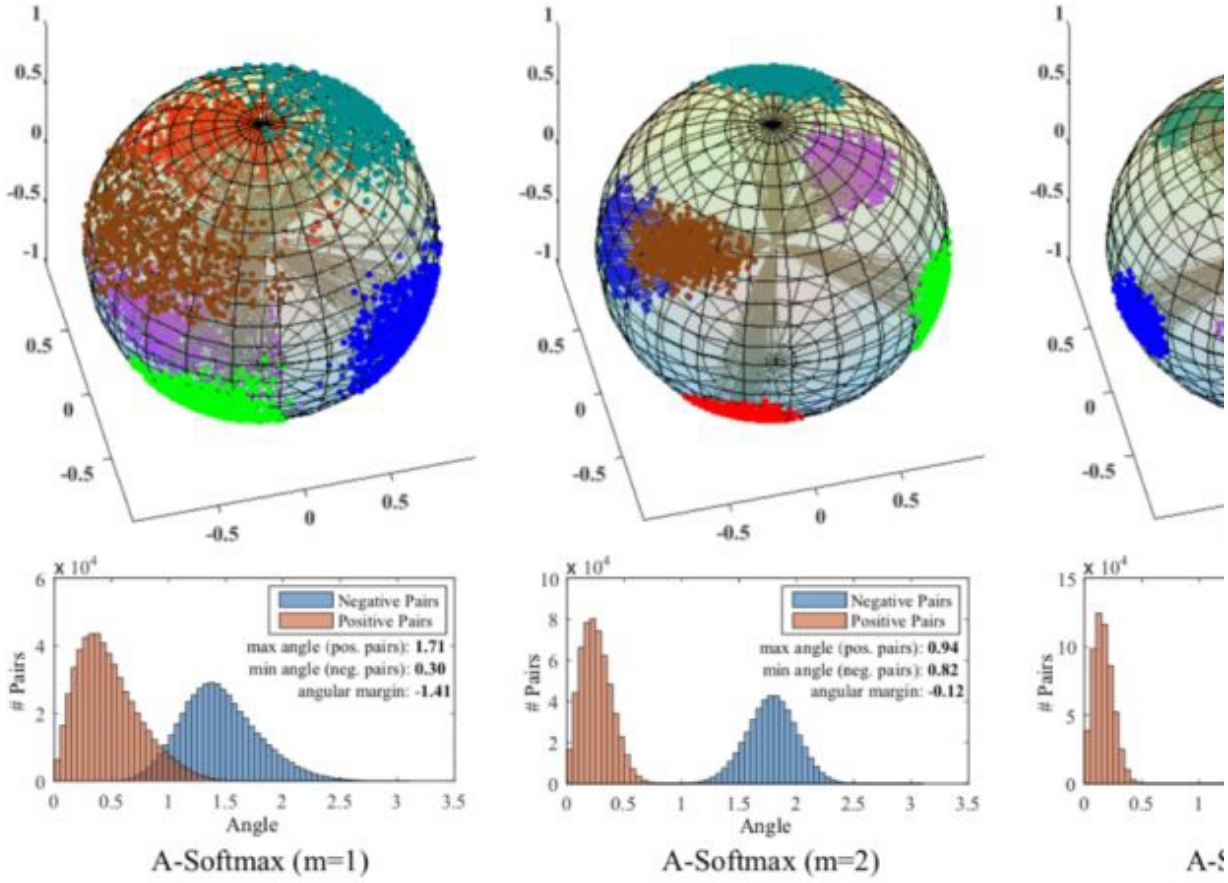
$\mathbf{w}_1$

和

$\mathbf{w}_2$

的夹角。不过可惜的是，这篇 paper 发表的比较早，在同时期 *[WeightNorm]* 才被发表出来，因此，在 *[L-Softmax]* 中并没有引入 Weight Normalization。2017 年 CVPR 的 *[SphereFace]* 在 L-Softmax 的基础上引入了 Weight Normalization。

*[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 loss 的具体形式是

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

，这个与 L2-Softmax 的差别仅仅就是把当前 sample 与所属类别的夹角

$\theta_{y_i,i}$

变成了

$m\theta_{y_i,i}$

。

但引入 margin 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 margin 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 margin 造成的。我们来分析一下，乘性 margin 到底带来的麻烦是什么：

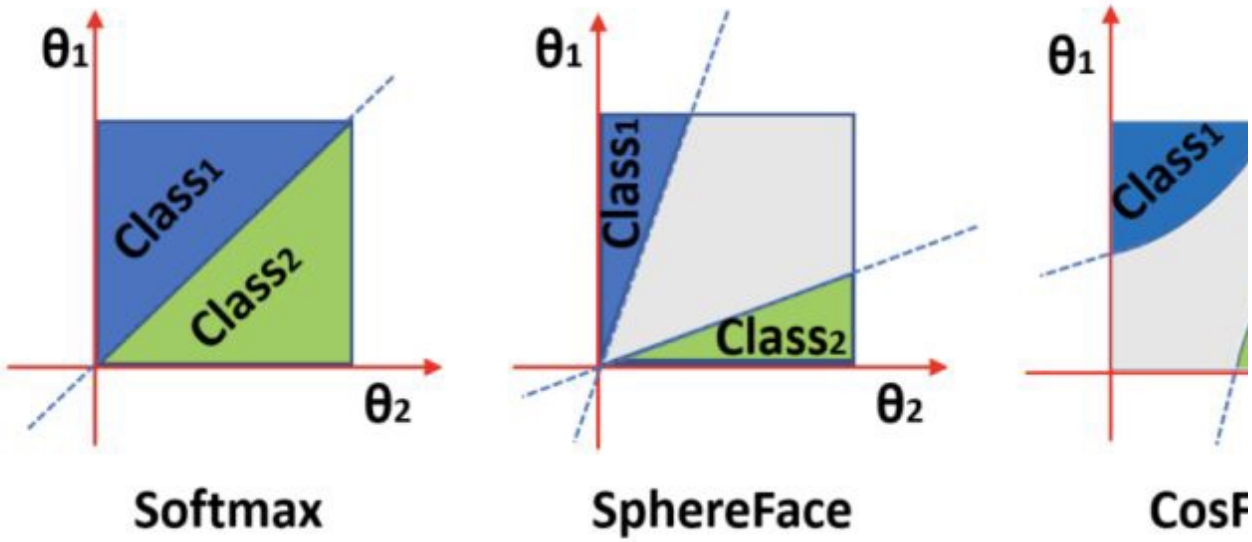
1. 第一点，乘性 margin 把 cos 函数的单调区间压小了，导致优化困难。对 $\cos(\theta_{y_i,i})$，在 处在区间 [0, π] 时，是一个单调函数，也就是说 落在这个区间里面的任何一个位置，网络都会朝着把 减小的方向优化。但加上乘性 margin m 后 的单调区间被压缩到了 $\left[0, \frac{\pi}{m}\right]$，那如果恰巧有一个 sample 的 落在了这个单调区间外，那网络就很难优化了；

2. 第二点，乘性 margin 所造成的 margin 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 class 之间的 angular decision margin $\frac{m-1}{m+1}\theta_{1,2}$，其中 是两个 class 的 weight 的夹角。这自然带来一个问题，如果这两个 class 本身挨得很近，那么他们的 margin 就小。特别是两个难以区分的 class，可能它们的 weight 挨得特别近，也就是 几乎接近 0，那么按照乘性 margin 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 margin 对易于混淆的 class 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 margin 的弊端自然就消失了。

同时期的另一篇 paper *[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 margin，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 margin 的方式做了很形象的对比，如下图所示。可以看出，*[ArcFace]* 提出的 margin 更符合“角度”margin 的概念，而 *[CosFace]* 或是 *[AM-Softmax]* 更符合 Cosine margin 的概念。




图片来自 *[Arcface]*

最后，我们总结一下加 margin 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 | $\frac{1}{N}\sum_i -log(\frac{e^{\|\mathbf{w}_{y_i}\|\cdot\|\mathbf{x}_i\|\cdot\cos(m\cdot\theta_{y_i,i})}}{e^{\|\mathbf{w}_{y_i}\|\cdot\|\mathbf{x}_i\|\cdot\cos(m\cdot\theta_{y_i,i})}+\sum_{k\neq y_i}e^{\|\mathbf{w}_k\|\cdot\|\mathbf{x}_i\|\cdot\cos\theta_{k,i}}})$ | $(\|W_1\|\cos m\theta_2)=0$ |
| 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

本文梳理了人脸识别领域近几年 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 metric 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，noisy sample 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 weight 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 margin 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。

▲长按关注我们

**麻烦给我一个在看！**