【论文解读】L-Softmax：用大间隔Softmax损失函数改进CNN

tomsheep

互联网行业 从业人员

参考论文：

- [1612.02295] Large-Margin Softmax Loss for Convolutional Neural Networks

辅助论文：

- Dimensionality Reduction by Learning an Invariant Mapping
- [1503.03832] FaceNet: A Unified Embedding for Face Recognition and Clustering

TL;DR

- **动机**：
  - 随着模型的加深和数据复杂度的增加，CNNs也面临着**过拟合（Overfitting）** 的问题，常见的手段有增大数据规模、正则化技术✦、数据增强、改进网络结构等。
  - 从损失函数✦角度出发，也可以缓解过拟合问题：要一种新的损失函数，既能保留Softmax损失的优点，又能显式地增强特征的判别性
  - 有研究者提出了**对比损失函数✦（Contrastive Loss）** 和**三元组损失函数（Triplet Loss）**。但它们需要精心设计的样本对或三元组，计算复杂度高，训练困难。
- **L-Softmax**损失函数（Large-Margin Softmax Loss）通过引入**可调节的角度间隔**，增强了特征的**类内紧凑性**和**类间可分性**。
- 通过调整参数m，可以控制学习目标的难度，避免过拟合，同时提升模型的泛化能力。
- **实验结果**：验证了L-Softmax损失在多个数据集上的有效性，在分类和人脸验证任务中都取得了显著的性能提升。

一、背景

1.1 卷积神经网络✦在视觉识别中的成功应用

近几年来，卷积神经网络（CNNs）在视觉识别领域取得了显著的成功，被广泛应用于各种任务，如：

- **物体识别**：如ImageNet大规模视觉识别挑战赛中，CNNs取得了优秀的成绩。
- **人脸验证**：如DeepFace、DeepID等模型实现了近乎人类水平的人脸识别。
- **手写数字识别**：经典的MNIST手写数字数据集上，CNNs的表现非常出色。

CNNs通过层次化的学习架构，以及卷积和池化操作，从局部到全局提取特征，具备强大的视觉表示能力。

1.2 过拟合问题及其应对策略

然而，随着模型的加深和数据复杂度的增加，CNNs也面临着**过拟合（Overfitting）** 的问题，即模型在训练数据上表现良好，但在未见过的数据上表现较差。为了解决过拟合问题，研究者们提出了多种策略：

- **大规模数据集**：使用更多的训练数据，如ImageNet。
- **正则化技术**：如Dropout，通过随机舍弃部分神经元，减少过拟合。
- **数据增强**：通过旋转、缩放等方式增加数据的多样性。
- **改进网络结构**：引入更深的网络，如VGGNet、ResNet。

1.3 增强特征的判别性

为进一步提高模型的泛化能力，研究人员开始关注**特征的判别性**，希望学习到的特征在类内更加紧凑，类间更加可分。具体而言：

- **类内紧凑性**：同一类别的特征应该聚集在一起，距离较近。
- **类间可分性**：不同类别的特征应该彼此远离，具有明显的间隔。

为实现这一目标，有研究者提出了**对比损失函数（Contrastive Loss）** 和**三元组损失函数（Triplet Loss）**。这些损失函数在训练过程中，分别使用样本对和样本三元组来增强特征的判别性。

1.4 Softmax损失的局限性

在CNNs中，**Softmax函数**结合**交叉熵损失**（**Cross-Entropy Loss**） 是最常用的监督组件。然而，传统的Softmax损失函数有以下局限性：

- **未显式鼓励特征的判别性**：Softmax损失主要关注样本的正确分类，而未对特征的类内紧凑性和类间可分性进行显式约束。
- **无法调整分类间隔**：Softmax损失无法控制类间的分类间隔，可能导致特征分布重叠，使得模型泛化能力受限。

二、动机

鉴于上述背景，我们需要一种新的损失函数，既能保留Softmax损失的优点，又能显式地增强特征的判别性。具体来说，我们的动机是：

- **解决Softmax损失的局限性**：通过改进Softmax损失，显式地鼓励类内紧凑性和类间可分性。
- **避免对比损失和三元组损失的复杂性**：对比损失和三元组损失需要精心设计的样本对或三元组，计算复杂度高，训练困难。
- **保持优化的便利性**：新的损失函数应当易于优化，能够与现有的优化方法（如随机梯度下降）无缝结合。

三、提出的大间隔Softmax损失函数（L-Softmax）

3.1 直觉及基本思想

在分类问题中，我们希望模型不仅能够正确分类样本，还能够确保不同类别之间具有**足够的间隔**，以增强模型的泛化能力。为此，我们引入了**角度间隔**的概念，具体包括以下两点：

- **角度度量相似性**：在神经网络的最后一层，全连接层的输出可以表示为特征向量与权重向量的内积，即通过角度（余弦相似度✦）来度量样本与分类器之间的相似性。
- **引入可调节的间隔**：通过将角度乘以一个常数m，使得分类决策更加严格，从而引入一个可调节的角度间隔。

举例说明：

假设我们有一个二分类问题，对于一个样本x，其对应的标签为类别1（$y_i = 1$），则原始的Softmax损失要求：

$$\|W_1\|\|x_i\|\cos\theta_1 > \|W_2\|\|x_i\|\cos\theta_2$$

即样本与类别1的权重向量的相似度大于与类别2的相似度。而我们希望通过引入一个常数m，使得分类条件更加严格：

$$\|W_1\|\|x_i\|\cos(m\theta_1) > \|W_2\|\|x_i\|\cos\theta_2$$

这意味着只有当样本与正确类别的角度满足更小的条件，它才能被正确分类，从而引入了一个角度间隔。

3.2 L-Softmax损失函数的数学定义

首先，我们回顾一下原始的Softmax损失函数。对于一个样本x_i，其标签为y_i，网络的最后一层输出f_i 通过一个全连接层✦和Softmax激活：

$$L = \frac{1}{N}\sum_{i=1}^{N} -\log\frac{e^{f_{y_i}}}{\sum_{j=1}^{K} e^{f_j}}$$

其中，$f_j = W_j^T x_i$ ，W_j 是第j 个类别的权重向量，K 是类别总数。

在L-Softmax损失函数中，我们对f_{y_i} 进行了修改，引入了角度间隔。具体地，L-Softmax损失函数定义为：

$$L = \frac{1}{N} \sum_{i=1}^{N} -\log \frac{e^{\|W_{y_i}\| \|x_i\| \psi(\theta_{y_i})}}{e^{\|W_{y_i}\| \|x_i\| \psi(\theta_{y_i})} + \sum_{j \neq y_i} e^{\|W_j\| \|x_i\| \cos(\theta_j)}}$$

其中：

- θ_j 是x_i 与W_j 之间的夹角，即：
  $\cos \theta_j = \frac{W_j^T x_i}{\|W_j\| \|x_i\|}$
- $\psi(\theta_{y_i})$ 是一个函数，用于引入角度间隔，定义为：
  $\psi(\theta_{y_i}) = (-1)^k \cos(m\theta_{y_i}) - 2k$ 其中，$k \in [0, m-1]$ ，使得$\theta_{y_i} \in [\frac{k\pi}{m}, \frac{(k+1)\pi}{m}]$ ，m 是一个整数，控制间隔的大小。

具体步骤

1. **计算每个类别的得分**：
   对于$j \neq y_i$ ：
   $f_j = \|W_j\| \|x_i\| \cos \theta_j$ 对于$j = y_i$ ：
   $f_{y_i} = \|W_{y_i}\| \|x_i\| \psi(\theta_{y_i})$
2. **计算损失**：
   $L = -\log \frac{e^{f_{y_i}}}{\sum_{j=1}^{K} e^{f_j}}$

3.3 几何解释

L-Softmax损失函数通过引入角度间隔，实现了对特征空间的重新约束，增强了特征的判别性。

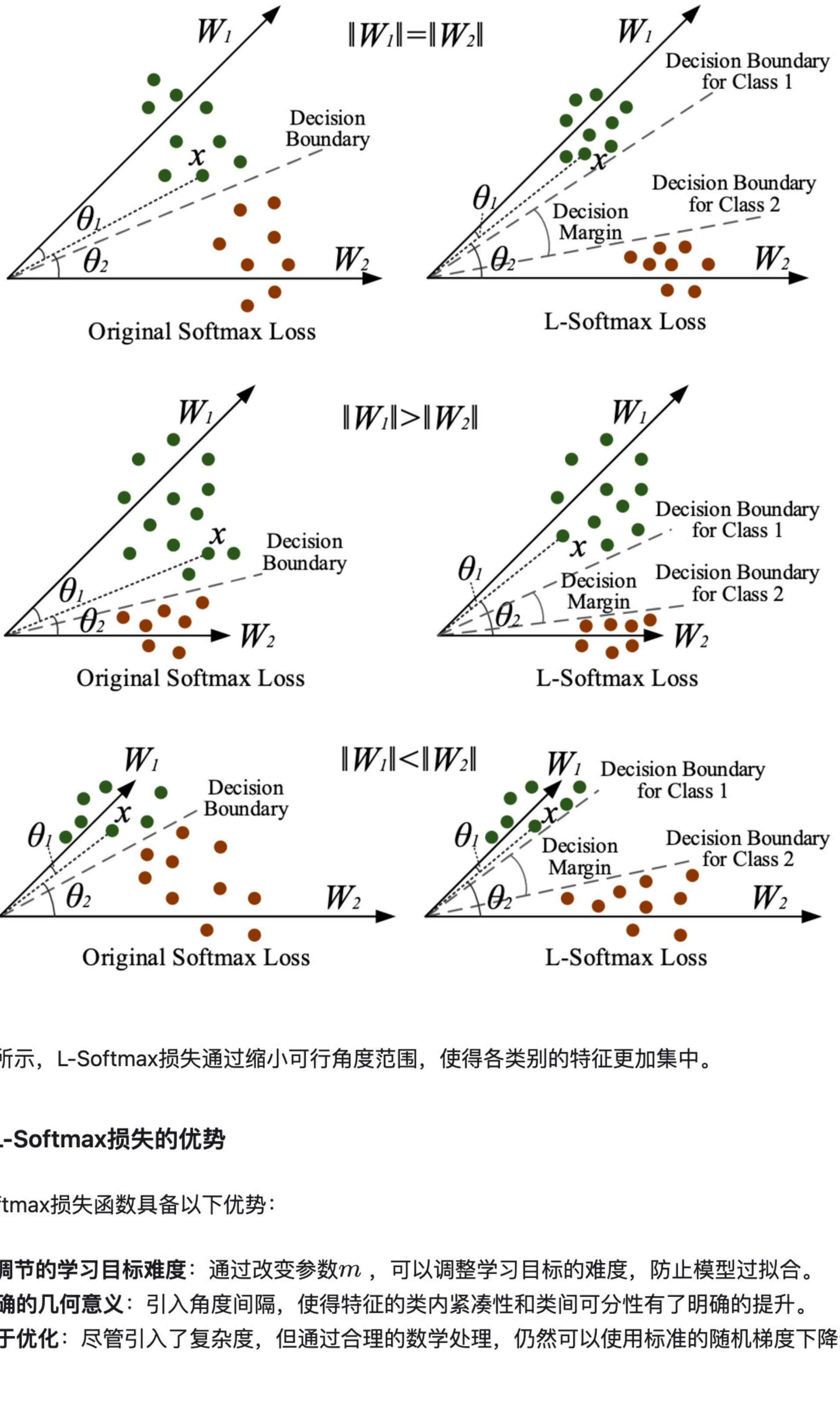
单独考虑两类的情况

假设有两个类别，权重向量分别为W_1 和W_2 ，且$\|W_1\| = \|W_2\|$ 。在特征空间中，分类边界由两者的角度决定。

- **原始Softmax损失**：
  对于一个样本x_i ，分类条件是$\theta_1 < \theta_2$ ，即x_i 与W_1 的夹角小于与W_2 的夹角。
- **L-Softmax损失**：
  引入m 后，分类条件变为$m\theta_1 < \theta_2$ ，即要求θ_1 更小，增加了分类的严格性。

间隔的扩大

通过引入m，实际效果是**缩小了同一类别的可行角度范围**，同时**扩大了不同类别之间的角度间隔**。这使得特征在类内更加紧凑，类间更加分离。



如图所示，L-Softmax损失通过缩小可行角度范围，使得各类别的特征更加集中。

3.4 L-Softmax损失的优势

L-Softmax损失函数具备以下优势：

1. **可调节的学习目标难度**：通过改变参数m，可以调整学习目标的难度，防止模型过拟合。
2. **明确的几何意义**：引入角度间隔，使得特征的类内紧凑性和类间可分性有了明确的提升。
3. **易于优化**：尽管引入了复杂度，但通过合理的数学处理，仍然可以使用标准的随机梯度下降进行

优化。

四、优化技巧

4.1 前向传播的计算

对于前向传播，需要计算$\cos(m\theta_{y_i})$，这可以通过多项式展开✦来实现，避免直接计算高次余弦函数。具体而言，利用如下公式：

$$\cos(m\theta) = 2\cos((m-1)\theta)\cos\theta - \cos((m-2)\theta)$$

通过递归和迭代，可以高效地计算$\cos(m\theta)$。

4.2 反向传播的梯度计算

在反向传播中，需要计算损失对参数的梯度，包括对特征x_i和权重W_j的梯度。通过链式法则✦，并结合对$\cos(m\theta_{y_i})$的导数，能够得到明确的梯度表达式。

4.3 训练策略

为了避免训练困难，特别是在m较大时，可以采用一些训练策略：

- **参数分段训练**：初始时使用较小的m，待模型收敛后，逐步增大m。
- **学习率调节**：适当降低学习率，防止梯度震荡。
- **正则化**：结合其他正则化手段，如批量归一化✦（Batch Normalization）和权重衰减✦（Weight Decay）。

五、实验结果

5.1 实验设置

为了验证L-Softmax损失的有效性，作者在以下数据集上进行了实验：

1. **MNIST**：手写数字识别✦数据集，共10类。
2. **CIFAR-10**：自然图像数据集✦，共10类。
3. **CIFAR-100**：自然图像数据集，共100类。
4. **LFW**：人脸验证数据集✦，用于评估特征的判别性。

网络结构

作者采用了统一的网络结构，仅在损失函数上进行对比。网络结构包括多个卷积层、池化层✦和全连接层，具体配置可参考论文中的表格。

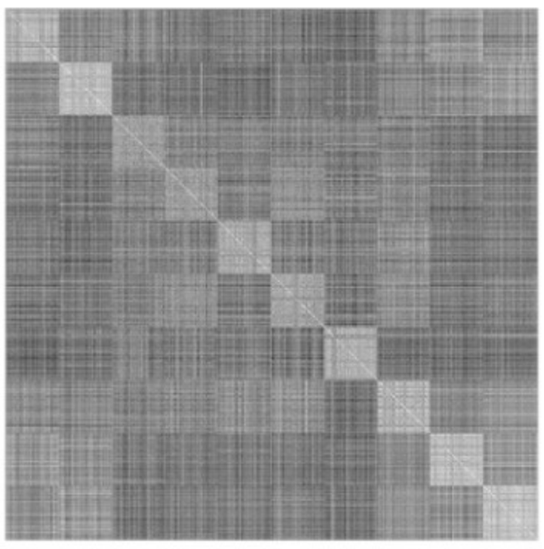
训练参数

- **优化方法**：随机梯度下降。
- **批量大小**：256。
- **学习率**：初始为0.1，根据训练进度逐步降低。
- **正则化**：使用PReLU激活函数✦和批量归一化，不使用Dropout。

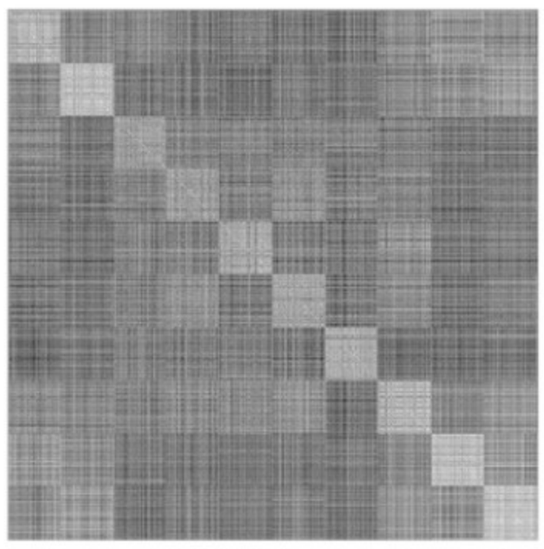
5.2 可视化分类结果

在MNIST数据集上，作者将特征向量降维到二维，并可视化了不同m 值下的特征分布。

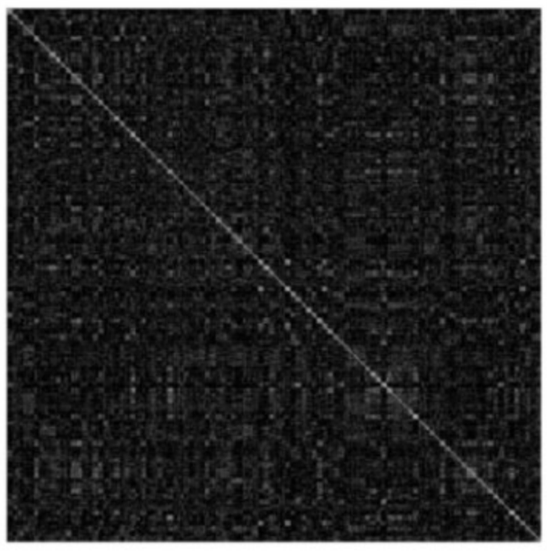
- $m = 1$ **（原始Softmax损失）**：
  特征分布较为分散，不同类别之间存在一定的重叠。
- $m = 2, 3, 4$ **（L-Softmax损失）**：
  随着m 的增加，特征分布逐渐集中，同类别样本聚集在一起，不同类别之间的间隔变大。



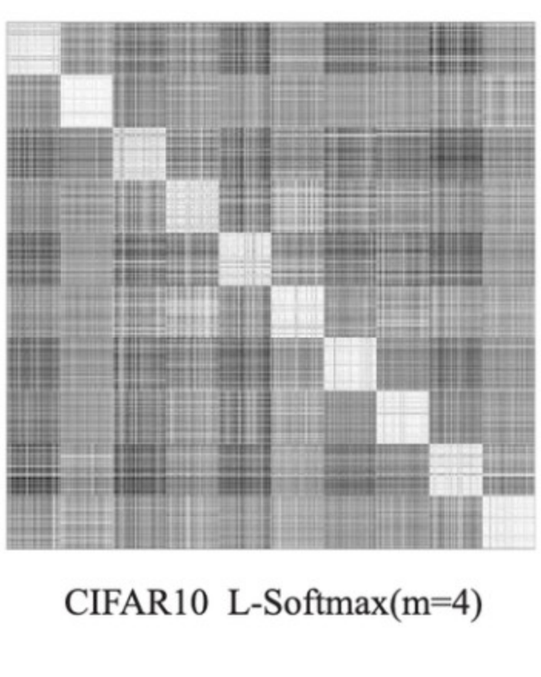
CIFAR10 Softmax



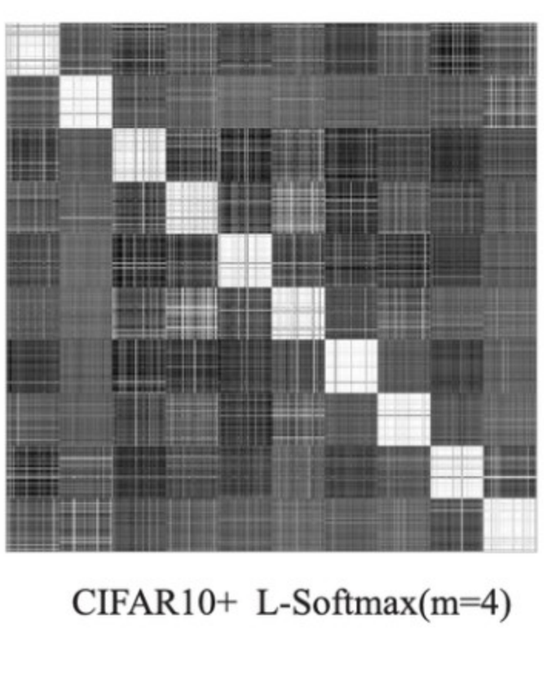
CIFAR10+ Softmax



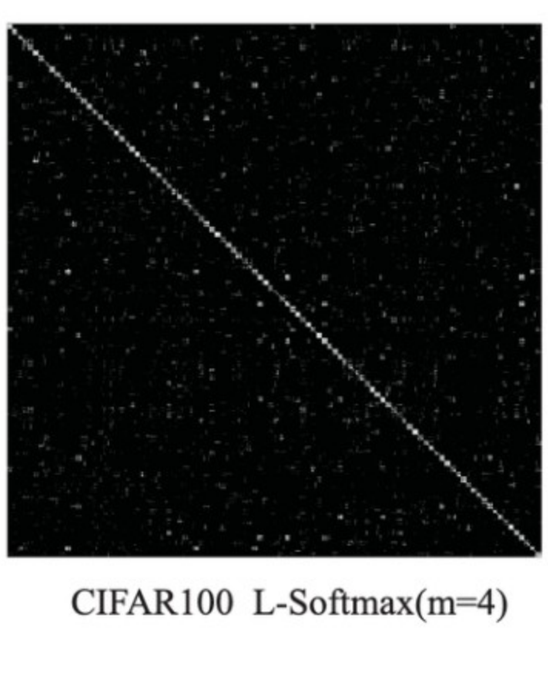
CIFAR100 Softmax



CIFAR10 L-Softmax(m=4)



CIFAR10+ L-Softmax(m=4)



CIFAR100 L-Softmax(m=4)

5.3 分类性能提升

MNIST数据集

在MNIST上，L-Softmax损失取得了更低的错误率，具体数值如下：

方法	错误率
Softmax	0.32%
L-Softmax (m=2\)	0.29%
L-Softmax (m=3\)	0.29%
L-Softmax (m=4\)	0.29%

CIFAR-10数据集

在CIFAR-10上，L-Softmax损失同样表现出优越性：

方法	错误率（无数据增强）	错误率（有数据增强）
Softmax	9.05%	6.50%
L-Softmax (m=2\)	7.73%	6.01%
L-Softmax (m=3\)	7.66%	5.94%
L-Softmax (m=4\)	7.58%	5.92%

CIFAR-100数据集

在更具挑战性的CIFAR-100数据集上，L-Softmax损失的优势更加明显：

方法	错误率
Softmax	32.74%
L-Softmax (m=2\)	29.95%
L-Softmax (m=3\)	29.87%
L-Softmax (m=4\)	29.53%

5.4 人脸验证性能提升

在LFW人脸验证数据集上，作者采用了公共可用的CASIA-WebFace数据集进行训练，结果显示：

方法	验证准确率
Softmax	96.53%
Softmax + 对比损失	97.31%
L-Softmax (m=2\)	97.81%
L-Softmax (m=3\)	98.27%
L-Softmax (m=4\)	98.71%

结果表明，L-Softmax损失显著提高了特征的判别性，与当前最先进的方法相比取得了竞争性的性能。

六、QA

1. 引入m，是否会增加训练的难度?

引入m 确实会增加学习目标的难度，因为模型需要满足更严格的分类条件。然而，这种增加的难度可以帮助模型避免过拟合，提升泛化能力。为了防止优化困难，可以采取以下策略：

- **逐步增加m**：从较小的m 开始训练，待模型收敛后，再逐步增大m 。
- **调整学习率**：适当降低学习率，保证模型的稳定训练。

2. L-Softmax损失会增加计算复杂度吗?

虽然L-Softmax损失引入了$\cos(m\theta)$ 的计算，但通过多项式展开和递归公式，可以高效计算而不显著增加计算复杂度。此外，网络的前向和反向传播过程仍然可以采用标准的优化方法。

3. 该方法适用于所有的分类任务吗?

L-Softmax损失主要适用于需要增强特征判别性的任务，特别是在类内紧凑性和类间可分性对性能有较大影响的情况下。如人脸识别、细粒度分类等任务都可以受益于L-Softmax损失。

4. 如何选择合适的m 值?

m 的选择需要根据具体任务和数据集进行实验。一般来说，m 取2到4之间即可取得明显效果。过大的m 可能导致收敛困难或训练时间过长，因此需要在优化难度和性能提升之间找到平衡。

5. 什么是对比损失函数（Contrastive Loss）?

对比损失函数是一种用于度量学习的损失函数，它通过比较成对的样本，推动模型学习到一个映射，使得：

- **同一类别的样本对**：在特征空间中彼此接近。
- **不同类别的样本对**：在特征空间中彼此远离，距离大于一个预设的阈值（称为**边际值**，margin）。

数学定义为：

$$L = \frac{1}{2N} \sum_{i=1}^{N} \left[y_i D^2 + (1 - y_i) \max(0, m - D)^2 \right]$$

其中：

- N 是样本对的数量。
- y_i 是样本对的标签，当样本对属于同一类别时，$y_i = 1$ ，否则$y_i = 0$ 。
- D 是样本对在特征空间中的欧氏距离：$D = \|f(x_i) - f(x_j)\|$ 。
- m 是边际值（margin），用于控制不同类别样本的最小距离。

假设我们有一个模型f ，用于将输入样本映射到特征空间。对于一个样本对(x_i, x_j) ：

- **如果x_i 和x_j 属于同一类别**（$y_i = 1$ ），损失函数为：
  $L_{same} = \frac{1}{2}D^2$ 模型被鼓励使得D 尽可能小，即特征距离越小越好。
- **如果x_i 和x_j 属于不同类别**（$y_i = 0$ ），损失函数为：
  $L_{diff} = \frac{1}{2}[\max(0, m - D)]^2$ 当$D \geq m$ 时，损失为零，表示不同类别的样本距离已经足够大；当$D < m$ 时，存在损失，模型被鼓励将样本拉远，直到距离大于m 。
- **优点**：能够直接优化特征空间的距离关系，增强类内紧凑性和类间可分性。
- **缺点**：在大型数据集上，样本对的数量为$O(N^2)$ ，计算复杂度高；需要精心选择**正负样本对**，避免不平衡和训练效率低下。

6. 什么是三元组损失函数（Triplet Loss）？

三元组损失函数通过比较样本三元组，推动模型学习到一个映射，使得：

- **锚点样本（Anchor）** 与**正样本（Positive）** 距离尽可能近。
- **锚点样本**与**负样本（Negative）** 距离尽可能远，且远离程度至少超过一个边际值（margin）。

数学定义为：

$$L = \frac{1}{N}\sum_{i=1}^{N}\left[\|f(x_i^a) - f(x_i^p)\|^2 - \|f(x_i^a) - f(x_i^n)\|^2 + m\right]_+$$

其中：

- N 是样本三元组的数量。
- x_i^a 是第i 个锚点样本。
- x_i^p 是与锚点属于同一类别的正样本。
- x_i^n 是与锚点属于不同类别的负样本。
- m 是边际值（margin）。
- $[z]_+ = \max(z, 0)$ 表示取非负值。

对于一个三元组(x_i^a, x_i^p, x_i^n) ：

- **正样本距离**：$D_{ap} = \|f(x_i^a) - f(x_i^p)\|$ 。
- **负样本距离**：$D_{an} = \|f(x_i^a) - f(x_i^n)\|$ 。

损失函数试图满足以下条件：

$$D_{ap} + m \leq D_{an}$$

即希望正样本距离加上边际值后，仍然小于负样本距离。

如果该条件不满足，即$D_{an} - D_{ap} < m$，则存在损失，模型被鼓励拉近锚点和正样本，拉远锚点和负样本。

- **优点**：直接优化了样本之间的相对距离关系，更加精确地控制特征空间的结构。
- **缺点**：可能需要大量的三元组，样本数量为$O(N^3)$；需要精心设计三元组选择策略（如在线挖掘困难样本），否则训练效率低下。

7. L-Softmax与对比损失和三元组损失的关系?

- **目的相同**：这三种损失函数都旨在增强模型学习到的特征的**类内紧凑性**和**类间可分性**，通过对特征空间施加更严格的约束，防止模型在训练集上过拟合提高模型的判别能力和泛化性能。
- **对比损失函数和三元组损失函数**是直接在特征空间中施加距离约束，需要计算样本之间的欧氏距离，需要精心设计样本对或三元组，计算复杂度较高。
- **L-Softmax损失函数**通过修改分类器的决策边界（引入可调节的角度间隔），在特征学习过程中施加更严格的分类条件，间接地使得特征空间中同类样本更集中，异类样本更分散。这样一来，保持了训练过程的简洁性，计算复杂度与原始Softmax接近，更易于在大规模数据集和深度模型中应用。