Differentiable Top-k Classification Learning

娄继猛

摘要

分类是机器学习和计算机视觉的核心学科之一。具有数百甚至数千个类别的分类问题的出现让 top-k 分类准确度成为一个重要的指标，这里，k 通常是一个正整数，例如 1 或 5，导致 top-1 或 top-5 训练目标。

平时我们所说的准确率其实就是 Top-1 准确率，而 Top-k 准确率则是用来计算预测结果中概率最大的前 K 个结果包含正确标签的占比，即 Top-k 准确率考虑的是预测结果中最有可能的 K 个结果是否包含真实标签。

在这项工作中，我们放宽了这个假设并同时针对多个 k 优化模型，而不是使用单个 k。利用可微排序和排序方面的最新进展，我们提出了一种可微的 top-k 交叉熵分类损失。这允许在训练网络的同时不仅考虑 top-1 预测，而且还考虑例如 top-2 和 top-5 预测。我们评估提出的损失函数，用于在最先进的架构上进行微调，以及从头开始训练。我们发现放松 k 不仅可以产生更好的 top-5 准确度，而且还可以提高 top-1 准确度。

关键词：difftop-k；classification learning

1 引言

分类是机器学习和计算机视觉的核心学科之一。数百甚至数千个类的分类问题的出现让 top-k 分类精度成为一个重要的度量标准，即 top-k 类中的一个必须是正确的类。

通常，对模型进行训练以优化 top-1 精度;top-5 等仅用于评估。Lapin 和 Berrada 等人所做的一些工作对这一观点提出了质疑，并提出了 top-k 损失，例如平滑的前 5 个边界损失。在存在额外标签噪声的情况下，这些方法已经证明了优于所建立的 top-1 softmax 交叉熵的鲁棒性。然而，在标准分类设置中，这些方法到目前为止还没有显示出优于所建立的 top-1softmax 交叉熵的改进。

在这项工作中，我们没有选择单一的 top-k 指标 (如 top-1 或 top-5) 来定义损失，而是建议从分布 PK 中指定 k，这可能依赖于特定数据点的置信度，也可能不依赖于类标签。分布 P_k 的例子有 [.5, 0, 0, 0, .5] (50% 的 top-1 和 50% 的 top-5)，[.1, 0, 0, 0,.9] (10% 的 top-1 和 90% 的 top-5)，和 [.2，.2，.2，.2，.2](从 1 到 5，每个 k 的 topk 值为 20%)。注意，当 k 是从分布中抽取时，这是无抽样的，因为我们可以以封闭形式计算期望值。

通常，给定神经网络返回的分数，softmax 会生成前 1 名的概率分布。Grover 和 Cuturi 等人提供了将其推广到由矩阵 P 表示的所有等级的概率分布的方法[1]。基于可微排序，提出了多个可微 top-k 算子。他们在可微 k 近邻算法、可微光束搜索、注意力机制和可微图像补丁选择中发现了应用。在这些领域，集成可微分 top-k 通过创建更自然的端到端学习设置，大大提高了结果表现。然而，到目前为止，还没有一个可微 top-k 算子被用于 k > 1 的 top-k 分类学习的神经网络损失。

在可微排序和排序方法的基础上，我们提出了一种新的可微 top-k 分类损失族，其中 k 来自概率分布。我们发现，我们的 top-k 损失不仅提高了 top-k 的准确性，而且还提高了多个学习任务的 top-1

的准确性。在 ciafar-100、ImageNet-1K 和 ImageNet-21K-P 数据集上使用四种可微排序和排名方法对论文的方法进行了验证评估[2]。使用 CIFAR-100，演示了 difftop-k 损失从头开始训练模型的能力。

2 相关工作

我们将相关工作分为三个大的部分: 导出和应用可微 top-k 算子的工作，一般使用排名和 top-k 训练目标的工作，以及呈现经典选择网络的工作。

2.1 可微 Top-k 算子

Grover 等人做了一个实验，他们使用 NeuralSort 可微 top-k 算子进行 kNN 学习[3]。Cuturi 和 Blondel 以及 Petersen 等人分别将其可微分排序和排序方法应用于 k = 1 的 top-k 监督[4]。

Xie 等人提出了一种基于最优传输和 Sinkhorn 算法的可微 top-k 算子。他们将他们的方法应用于 k-近邻学习 (kNN)，带排序软 top-k 的微分光束搜索，以及用于机器翻译的 top-k 注意。

Cordonnier 等人使用摄动优化器推导出可微的 top-k 算子，他们将其用于可微的图像补丁选择。Lee 等人提出使用 NeuralSort 作为可微的 top-k 算子，为推荐系统生成可微的排名指标。

Goyal 等人在 2018 年提出了一种用于可微光束搜索的连续 top-k 算子。Pietruszka 等人在 2020 年提出了可微连续对半 top-k 算子来近似归一化倒角余弦相似度。

2.2 排序和 Top-k 训练目标

Fan 等人 (2017) 提出了“平均 top-k”损失，这是一种对训练数据集的 k 个最大个体损失进行平均的总损失。他们将这种总损失应用于支持向量机进行分类任务。注意，这不是这个工作意义上的可微顶 k 损失。相反，top-k 是不可微的，用于决定哪些数据点的损失将汇总为损失。

Lapin 等人提出了多类支持向量机的松弛 top-k 代理误差函数[5]。受到学习排序损失的启发，他们提出了 top-k 校准、top-k 铰链损失、top-k 熵损失以及截断 top-k 熵损失。他们将他们的方法应用到多类 svm 中，并通过随机双坐标上升 (SDCA) 进行学习。

Berrada 等人在这些思想的基础上提出了用于深度 top-k 分类的平滑损失函数。

top-k 损失在 CIFAR-100 和 ImageNet1K 任务中表现良好。虽然与强 Softmax 交叉熵基准相比，他们的方法并没有提高原始数据集上的性能，但在标签噪声和数据集子集的设置中，他们提高了分类精度。具体来说，当 CIFAR-100 上的标签噪声为 20% 或更高时，它们可以提高 top-1 和 top-5 的精度，对于 ImageNet1K 的子集，它们可以提高 top-5 的精度，最高可达 50%。

然而，与 Berrada 等人相比，我们的方法提高了未修改设置下的分类精度。在我们的实验中，对于 k 是一个具体整数且不是从分布中抽取的特殊情况，我们提供了与平滑 top-k 代理损失的比较。

Y ang 和 Koyejo 提供了 topk 替代损失的理论分析，并提出了一种新的替代 top-k 损失，他们在合成数据实验中对其进行了评估。一个相关的思想是集值分类，其中预测一组标签。论文参考的广泛概述。我们注意到，我们的目标不是预测一组标签，而是为每个类返回一个对应于排名的分数，其中只有一个类可以对应于真值。

2.3 选择网络

先前的选择网络都是基于经典的分治排序网络，递归地排序子序列并合并它们[6]。在选择网络中，在合并过程中，只有前 k 个元素被合并，而不是完整的 (排序的) 子序列。与之前的工作相比，我们提

出了一类新的选择网络，它实现了更紧密的边界 (对于 k « N)，并放松它们。

3 本文方法

3.1 本文方法概述

在本节中，我们首先介绍我们的目标，详细阐述其精确公式，然后建立可微排序原则，以有效地近似目标。图 1 还给出了损耗体系结构的可视化概述。top-k 学习的目标是将学习标准从只接受精确的 (top-1) 预测扩展到接受 k 个预测，其中正确的类别必须要有。在其一般形式中，对于 top-k 学习，每个应用程序、类、数据点或其组合的 k 可能不同。

例如，在一种情况下，人们可能希望对 5 个预测进行排名，并分配一个取决于这些排名预测中真实类的排名的分数，而在另一种情况中，人们可能想要获得 5 个预测，但不关心它们的顺序。

在另一种情况下，比如：图像分类，人们可能希望对“人”的图像执行前 top-1 的精度，但对“动物”超类则接受前 top-3 精度，因为它可能在类标签中有更多的歧义。

提出的架构概述: CNN 预测图像的分数，然后通过可微分排名算法对图像进行排名，返回矩阵 P 中每个排名的概率分布。该分布的行对应排名，列对应各自的类。在本例中，我们使用 50% top-1 和 50% top-2 的损失，即 P_K =[.5, .5, 0, 0, 0]。这里，第 k 个值指的是第 k 个分量，如果预测在第 1 到第 k 个分量中的任意一个，则满足第 k 个分量。因此，不同等级的权重可以通过累积和计算，并为 [1, .5, 0, 0, 0]。相应的 P 行加权和产生概率分布 P，然后可用于交叉熵损失。

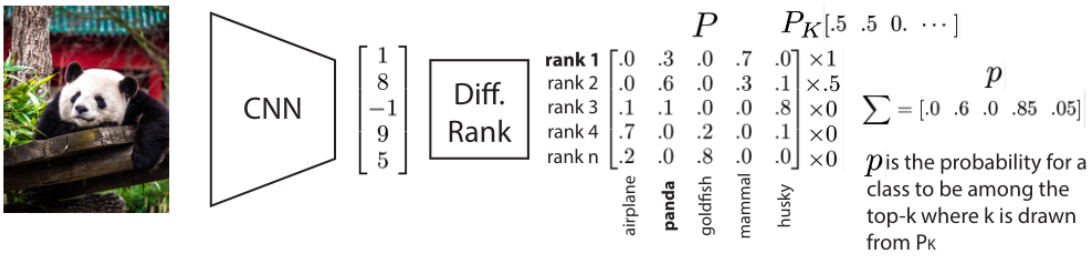

图 1: 架构概述图

3.2 Top-k 概率矩阵

所讨论的可微分排序算法产生大小为 n × n 的宽松排列矩阵。然而，对于 top-k 分类学习，我们只需要考虑排名最高的 k 个类的前 k 行。这里 k 是目标考虑的最大 k，即 P_K(k) > 0。当 n » K 时，生成一个 K × n 矩阵而不是 n × n 矩阵要快得多。

对于 NeuralSort 和 SoftSort，可以简单地只计算顶部行，因为算法是按行定义的。

对于可微 Sinkhorn 排序算法，不能直接提高运行时间，因为在每次 Sinkhorn 迭代中都需要完整的矩阵。Xie 等人提出了一个基于 sinkhorn 的可微分 top-k 算子，它计算了一个 2 × n 矩阵，其中第一行对应于 top-k 元素，第二行对应于其余元素[7]。然而，这种公式不产生 P，也不区分 top-k 元素之间的位置，因此我们使用 Cuturi 等人的 SinkhornSort 算法[8]。

对于可微排序网络，通过双向求值可以将代价从 $O(n^2log^2(n))$ 降低到 $O(nklog^2(n))$。在这里，重要的是要注意得到 P 的乘法的形状和顺序。由于我们只需要考虑排序网络的最后一层之后的前 k 位元素，我们可以省略最后一层 (t 层) 的置换矩阵的所有剩余行，因此它的大小仅为 (k × n)。

$$\underbrace{(k \times n)}_{P} = \underbrace{(k \times n)}_{layer \times t}\underbrace{(n \times n)}_{layer \times t-1}...\underbrace{(n \times n)}_{layer \times t}$$

注意，在排序网络的执行过程中，P 通常是从第 1 层到第 t 层计算，即从右到左。如果我们按照这个顺序计算，我们只会只在最后一层节省一小部分的计算成本。因此，我们建议执行可微排序网络，保存填充 (稀疏)n × n 层排列矩阵的值，并在从后到前的第二次传递中计算 P，即从 t 层到第 1 层，或在以上公式中从左到右。这允许使用密集的 k × n 矩阵和稀疏的 n × n 矩阵执行 t 次密集-稀疏矩阵乘法，而不是密集的 n × n 和稀疏的 n × n 矩阵。这样，我们将渐近复杂度从 $O(n^2log^2(n))$ 降低到 $O(nklog^2(n))$。

3.3 可微 Top-k 网络

由于 top-k 分类学习只需要松弛排列矩阵的 top-k 行，因此可以通过减少可微层和比较器的数量来提高可微排序网络计算 top-k 概率分布的效率。

因此，我们提出了可微的 top-k 网络，它放松了选择网络，类似于可微的排序网络放松了排序网络。选择网络是指从 n 个元素中只选择前 k 个元素的网络。

我们提出了分割器选择网络 (SSN)，这是一种新型的选择网络，它只需要 O(log n) 层 (而不是排序网络的 O(log2 n) 层)，它可以用可微分的 top-k 网络更高效进行 top-k 监督，并减少了误差。ssn 遵循这样的思想: 输入被分成局部排序的子列表，然后所有不在全局 top-k 候选列表中的连线都可以被删除。

例如，对于 n = 1024, k = 5, ssn 只需要 22 层，而最好的前选网络需要 34 层，完全排序 (对于二进制网络) 甚至需要 55 层。对于 n = 10450, k = 5(即 ImageNet-21K-P)，snn 需要 27 层，最好的前一层需要 50 层，完全排序需要 105 层。此外，与二进制排序网络相比，ssn 层的计算成本更低。总之，可微 top-k 网络的贡献有两个方面: 首先，我们提出了一种需要更少层的新型选择网络。其次，我们放松了与可微排序网络类似的选择网络。

3.4 实现细节

尽管有这些性能改进，但对于大量的类，计算可微排序操作仍然需要相当多的计算工作。特别是当待排序元素的数量 n = 1 000 (ImageNet-1K) 或 n > 10000 (ImageNet21K-P) 时，可微排序算子可以主导整体计算成本。此外，对于大量 n 个要排序的元素，可微排序算子的性能会下降，因为对更多元素进行差分排序自然会引入更大的误差。因此，我们通过只考虑对于每个输入那些得分在前 m 分中的类来减少要进行可微排序的输出数量。为此，如果必要的话，我们通过用 ground truth 类替换 top-m 分数中最低的一个，以确保 ground truth 类位于那些得分最高的 m 类中。对于 n = 1000，我们选择 m = 16，对于 n > 10 000，我们选择 m = 50。我们发现这极大地提高了训练效果。

因为可微排序算子 (由于它们的性质是可微的) 只是硬排序算子的近似值，它们各自都有自己的特征和不一致性。因此，对于从头开始训练模型，我们用常规的 softmax 替换损失的 top-1 分量，它具有更好和更一致的行为。

如果可微排序算子行为不一致，这将引导其他损失。为了避免 top-k 分量影响 softmax 分量，并避免 p 中的概率大于 1，我们可以将交叉熵分离为 softmax 交叉熵 (sm，对于 top1 分量) 和 top-k 交叉熵 (top -k，对于 top-k≥2 分量) 的混合，如下所示:

$$L_{\text{sm+top}-k}(X, y) = P_K(1) \cdot \text{SoftmaxCELoss}\left(f_\Theta(X), y\right)$$
$$-(1 - P_K(1)) \cdot \log\left(\sum_{k=2}^{n} P_K(k)\left(\sum_{m=1}^{k} \boldsymbol{P}_{m,y}\left(f_\Theta(X)\right)\right)\right)$$

4 复现细节

4.1 与已有开源代码对比

原论文暂时并未提供相关代码，核心算法需要自己查阅相关文献或请教作者动手实现。本报告主要工作包括：从头开始在 CIFAR-100 上训练 ResNet18、代码实现 TopK 交叉熵损失，以及 difftopk 算法等。本论文基于可微排序和排序方法，推导出了一个新的 top-k 交叉熵损失族，并放宽了固定 k 的假设，通过实验表明：top-k 损失不仅提高了 top-k 准确度，而且还提高了多个学习任务的 top-1 准确度！

4.2 实验环境搭建

1. 安装 difftopk 包

pip install difftopk

2. 安装稀疏计算相关包为了以稀疏方式评估可微 topk 运算符的功能，必须安装包火炬稀疏。

pip install torch-scatter torch-sparse -f https://data.pyg.org/whl/torch-1.13.0+cpu.html

3. 安装 numpy 和 pytorch 等相关包 pip install boto3 numpy requests scikit-learn tqdm

pip install torch==1.13.0+cu116 torchvision==0.14.0+cu116 -f https://download.pytorch.org/whl/torch_stable.h

4. 从头开始完整安装举例 virtualenv -p python3 .env_topk

. .env_topk/bin/activate

pip install boto3 numpy requests scikit-learn tqdm

pip install torch==1.13.0+cu116 torchvision==0.14.0+cu116 -f https://download.pytorch.org/whl/torch_stable.h

pip install diffsort

optional for torch-sparse

FORCE_CUDA=1 pip install –no-cache-dir torch-scatter torch-sparse -f https://data.pyg.org/whl/torch-1.13.0+cu116.html

pip install .

5. 实验参数设置：本实验参数较多，以下是部分核心参数

Tables 1+4:

python experiments/train_cifar100.py

–method softmax_cross_entropy

–method bitonic –distribution logistic_phi –inverse_temperature .5 –art_lambda .5

–method splitter_selection –distribution logistic_phi –inverse_temperature .5 –art_lambda .5

–method neuralsort –inverse_temperature 0.0625

–method softsort –inverse_temperature 0.0625

–method smooth_hard_topk –inverse_temperature 1.

–p_k 1. 0. 0. 0. 0.

–p_k 0. 0. 0. 0. 1.

–p_k .5 0. 0. 0. .5

–p_k .2 0. 0. 0. .75

–p_k .1 0. 0. 0. .9

–p_k .2 .2 .2 .2 .2

Examples:

python experiments/train_cifar100.py –method softmax_cross_entropy –p_k 1. 0. 0. 0. 0.

python experiments/train_cifar100.py –method splitter_selection –distribution logistic_phi –inverse$_{t}emperatur$
$-art\_lambda.5 - -p\_k.2.2.2.2.2$

4.3 创新点

在本文工作中，提出指定 k 从分布 Pk 中提取，而不是选择单个 top-k 度量（如 top-1 或 top-5）来定义损失，这可能取决于或可能不取决于特定数据点的置信度或类别标签。

1）基于可微排序和排序方法，推导出了一个新的 top-k 交叉熵损失族，并放宽了固定 k 的假设。

2）发现 top-k 损失不仅提高了 top-k 准确度，而且还提高了多个学习任务的 top-1 准确度。

5 实验结果分析

下面的表 1 描述了 CIFAR-100 从头开始训练 ResNet18 的结果。指标为 2 粒种子的平均精度 Top-1|Top-5。其中，Ours 为作者论文结果。My 部分为本报告复现结果。我们发现放松 k 不仅可以产生更好的 top-5 准确度，而且还可以提高 top-1 准确度。

Method	P_k	CIFAR-100
Baselines		
Softmax	([1, 0, 0, 0, 0])	61.27 | 85.31
Smooth top-k loss ($\star$)	([0, 0, 0, 0, 1])	53.07 |85.23
Top-5 NeuralSort	[0, 0, 0, 0, 1]	22.58 | 84.41
Top-5 SoftSort	[0, 0, 0, 0, 1]	1.01 | 5.09
Top-5 SinkhornSort	[0, 0, 0, 0, 1]	55.62 |87.04
Top-5 DiffSortNets	[0, 0, 0, 0, 1]	52.81 | 84.21
Ours		
Top-k NeuralSort	[.2, .2, .2, .2, .2]	61.46 | 86.03
Top-k SoftSort	[.2, .2, .2, .2, .2]	61.53 | 82.39
Top-k SinkhornSort	[.2, .2, .2, .2, .2]	61.89 |86.94
Top-k DiffSortNets	[.2, .2, .2, .2, .2]	62.00 | 86.73
My		
Top-k NeuralSort	[.2, .2, .2, .2, .2]	59.42 | 83.03
Top-k SoftSort	[.2, .2, .2, .2, .2]	63.54 | 80.17
Top-k SinkhornSort	[.2, .2, .2, .2, .2]	54.89 |82.34
Top-k DiffSortNets	[.2, .2, .2, .2, .2]	55.00 | 84.60

表 1: CIFAR-100 从头开始训练 ResNet18 的结果。指标为 2 粒种子的平均精度 Top-1|Top-5

6 总结与展望

分类是机器学习和计算机视觉的核心学科之一。具有数百甚至数千个类别的分类问题的出现让 top-k 分类准确度成为一个重要的指标，这里，k 通常是一个正整数，例如 1 或 5，导致 top-1 或 top-5 训练目标。

数百甚至数千个类的分类问题的出现，使得 top-k 分类精度成为一个重要的度量，即 top-k 类之一必须是正确的类。通常，对模型进行训练以优化最高精度；top 5 等仅用于评估。

平时我们所说的准确率其实就是 Top-1 准确率，而 Top-k 准确率则是用来计算预测结果中概率最大的前 K 个结果包含正确标签的占比，即 Top-k 准确率考虑的是预测结果中最有可能的 K 个结果是否包含真实标签。

在本论文中，我们放宽了这个假设并同时针对多个 k 优化模型，而不是使用单个 k，提出了一种可微的 top-k 交叉熵分类损失。这允许在训练网络的同时不仅考虑 top-1 预测，而且还考虑例如 top-2 和 top-5 预测。我们评估提出的损失函数，使用 CIFAR-100 从头开始训练 ResNet18 的结果。我们发现放松 k 不仅可以产生更好的 top-5 准确度，而且还可以提高 top-1 准确度。

通过本次前沿技术课程的学习，有效地磨练了我的科研态度和生活心性。感恩廖好老师的热心选材以及认真督导，老师让我懂得以后无论做科研还是做工作汇报，PPT 汇报上的内容都尽量要以图片为主，而尽量避免大篇幅文字内容的出现。图片部分可以是多个图的一个集成，优美大气、栩栩如生的图片不仅能够极大地提高听众地学习兴趣，于此同时这也能够体现出我们胸有成竹的专业实力！感恩所有帮助过我的老师以及小伙伴，真心祝愿未来的我们都能够有一个更为美好的未来！

参考文献

[1] GROVER A, WANG E, ZWEIG A, et al. Stochastic optimization of sorting networks via continuous relaxations[J]. arXiv preprint arXiv:1903.08850, 2019.

[2] KRIZHEVSKY A, HINTON G, et al. Learning multiple layers of features from tiny images[J]., 2009.

[3] BLONDEL M, TEBOUL O, BERTHET Q, et al. Fast differentiable sorting and ranking[C]//International Conference on Machine Learning. 2020: 950-959.

[4] PETERSEN F, BORGELT C, KUEHNE H, et al. Learning with algorithmic supervision via continuous relaxations[J]. Advances in Neural Information Processing Systems, 2021, 34: 16520-16531.

[5] LAPIN M, HEIN M, SCHIELE B. Loss functions for top-k error: Analysis and insights[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 1468-1477.

[6] WAH B W, CHEN K L. A partitioning approach to the design of selection networks[J]. IEEE transactions on computers, 1984, 100(3): 261-268.

[7] XIE Y, DAI H, CHEN M, et al. Differentiable top-k with optimal transport[J]. Advances in Neural Information Processing Systems, 2020, 33: 20520-20531.

[8] CUTURI M, TEBOUL O, VERT J P. Differentiable ranking and sorting using optimal transport[J]. Advances in neural information processing systems, 2019, 32.