

# 基于密度加权原型网络的小样本学习算法

华超,刘向阳

(河海大学理学院,江苏南京 211100)

**摘要:**小样本学习主要研究从少量的样本快速学习和归纳,原型网络对该问题提出了解决方法。原型网络算法将样本通过网络框架嵌入到低维空间,在低维空间中将支持集的每个类别的样本均值作为各自的原型,计算查询集样本到每个类别原型的距离,进而得到损失函数,通过迭代更新损失函数来优化网络框架,使得经过网络框架后的样本类内的差异变小,类间的差异变大。在计算每个类别原型的过程中,将每个类的样本均值作为各自的原型,由于每个类别的样本数较少,在计算原型时会有不确定性,当样本中存在离群点时,会使计算出的原型偏离真实的原型,在判别新样本属于哪一类时,很容易产生误差。针对该问题,提出密度加权原型网络,利用密度加权算法计算每个类别中样本的密度,对密度大的样本赋予较大的权值,密度小的样本赋予较小的权值,进行加权得到原型,来缓解以上的缺陷。在 miniImageNet 数据集上设置 5-way 5-shot、20-way 5-shot 实验,结果表明所提出密度加权原型网络相对于原型网络算法在识别正确率方面有所提升。

**关键词:**小样本学习;情节;密度加权;临界距离;距离矩阵

中图分类号:TP391

文献标识码:A

文章编号:1673-629X(2022)09-0008-06

doi:10.3969/j.issn.1673-629X.2022.09.002

## Few-shot Learning Based on Density-weighted Prototypical Network

HUA Chao, LIU Xiang-yang

(School of Science, Hohai University, Nanjing 211100, China)

**Abstract:** Few-shot learning mainly studies fast learning and induction from a small number of samples, and the prototype network proposes a solution to this problem. The prototypical network algorithm embeds the samples into the low-dimensional space through the network framework. In the low-dimensional space, the sample mean of each category from the support set is used as the respective prototype, and the distance between the query set sample and the prototype of each category is calculated, and then the loss function is obtained. The network framework is optimized by iterating and updating the loss function, so that the intra-class difference in the sample through the network framework become smaller and the inter-class difference become larger. In the process of calculating the prototype of each category, the prototype network uses the sample average of each category as its own prototype. Since the number of samples in each category is small, there will be uncertainty when calculating the prototype. When there are outliers in the sample, the calculated prototype will deviate from the real prototype. When judging which category the new sample belongs to, it is easy to produce errors. In response to this problem, we propose a density-weighted algorithm, which uses a density-weighted algorithm to calculate the density of samples in each category, to assign larger weights to samples with high density, and to assign smaller weights to samples with low density to alleviate the above shortcomings. The 5-way 5-shot and 20-way 5-shot experiments are set on the miniImageNet dataset. The results show that the proposed density-weighted prototype network has improved recognition accuracy compared to the prototype network algorithm.

**Key words:** few-shot learning; episodes; density weighting; critical distance; distance matrix

## 0 引言

随着计算机视觉的发展,深度学习的出现,各种深度网络在处理数据量大的数据集进行图像识别、目标检测、自然语言处理等任务时可以取得较好的结果,但

是深度学习为此而付出的成本却比较大。在进行深度学习时,为了得到较好的网络模型参数,需要大量有标签的数据和时间来训练网络,而在现实生活中获取大量有标签的数据成本很大,因此需要研究其他的方法

收稿日期:2021-09-23

修回日期:2022-01-25

基金项目:云南省重大科技专项计划项目资助(202002AE090010)

作者简介:华超(1997-),男,硕士研究生,研究方向为小样本学习;刘向阳,博士,副教授,研究方向为机器学习。

来解决这些问题。人类能够在少许的样本中学习并识别新的对象类别,为了让计算机也具备这种能力,提出了小样本学习。

Wang Yaqing 等<sup>[1]</sup>将小样本学习方法概括为以下三类:第一类是通过先验知识增强数据集,该方法是通过先验知识将少量有标签或无标签的数据集,小样本训练集和相似的数据集转换成足够的训练数据集,例如 E. D. Cubuk 等提出的从数据中学习增强策略<sup>[2]</sup>, J. Wei 等提出的用于提高文本分类任务性能的简单数据增强技术<sup>[3]</sup>等;第二类是通过先验知识限制假设空间,该方法是通过先验知识将样本假设空间约束到较小的假设空间来进行学习,从而更快地找到较好的结果,例如 Oriol Vinyals 等提出的匹配网络<sup>[4]</sup>, Snell J 等提出的原型网络<sup>[5]</sup>和 F. Sung 等提出的相关网络<sup>[6]</sup>等;第三类是通过先验知识在假设空间中改变搜索策略,该策略将从相关任务中学习的预训练模型进行较好的初始化,通过几次迭代适应数据集,例如 C Finn 等提出的与模型无关的快速适应深度网络元学习<sup>[7]</sup>和与模型无关的概率元学习<sup>[8]</sup>和 L. Bottou 等提出的大规模机器学习的优化方法<sup>[9]</sup>等。

Snell J 等提出的原型网络属于第二类,是基于嵌入学习的小样本学习的方法。该方法首先通过卷积神经网络将样本映射到更低维的嵌入空间,然后分别将每个类别的样本均值作为各自的原型,当遇到新样本时,计算该样本到每个原型的距离,最后将该样本判给距离原型最小距离的类别。原型网络算法简单地将每个类别的样本均值作为各自的原型,会带来一些缺点。第一,将每个类的样本均值作为各自的原型,从而忽略样本本身的特性,从而使得原型不具有代表性,在判别新样本属于哪一类时,很容易产生误差。第二,由于每个类的样本数较少,在计算原型时会有不确定性,当样本中存在离群点时,会使计算的原型偏离真实的原型。为了缓解以上的缺陷,该文对计算原型的方法提出了改进。

对于某一类的样本而言,原型网络算法利用均值计算原型,其本质是对每个样本赋予相同的权值,然后进行加权。而每个样本的代表性是不一样的,因此该文对每个样本赋予不同的权值。首先考虑每个样本在某固定范围内的样本个数是不同的,即样本在某固定范围内的样本个数越多,则该样本的密度越大,即该样本就越具有代表性;而样本在某固定范围内的样本个数越少,则该样本的密度越小,即该样本就越不具有代表性。该文给予代表性大的样本赋予更大的权值,给予代表性小的样本赋予更小的权值。基于上述的策略,提出了密度加权原型网络算法,为了将密度加权的思想更加生动地表示出来,绘制了二维的图像来表示。

假设样本点的分布如图 1 所示,样本点中存在一个偏离总体的样本点,为了减小该样本点对计算原型的影响,首先计算出每个样本点在固定的范围内的样本数,样本点在固定范围内的样本数越多,在图中以样本点为中心的范围标识的颜色越深,则样本点的密集就越大,赋予它较大的权值;而对于离群点,在图中以该样本点为中心的范围标识的颜色越浅,则样本的密集就越小,赋予它较小的权值,利用密度加权计算得到原型更加接近真实的原型,相对于直接计算所有样本点的均值作为原型更优。详细的理论在后面进行说明。

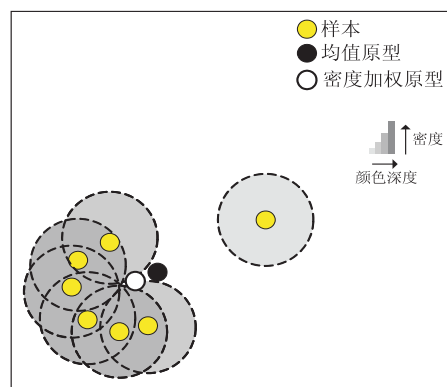


图 1 密度加权结构

## 1 原型网络

### 1.1 数据集分割

在原型网络算法中定义了情节 (episodes) 的方式,首先将数据集划分为训练集、验证集和测试集,其中训练集、验证集和测试集类别不相交,然后对它们进行进一步划分,从中选取  $N$  个类别  $K$  样本 ( $N$ -way  $K$ -shot) 称为一个情节。从划分好的数据集中抽取  $N$  个类别,每个类别中选取  $m$  个样本作为支持集  $S = \{(x_1, y_1), \dots, (x_{m \times N}, y_{m \times N}) \mid x_i \in R^{H \times W \times C}, y_i \in \{1, \dots, N\}\}$ ,从  $N$  个类别中剩余的样本中选取  $n$  个样本作为查询集  $Q = \{(x_1, y_1) \dots (x_{n \times N}, y_{n \times N}) \mid x_i \in R^{H \times W \times C}, y_i \in \{1, \dots, N\}\}$ 。

### 1.2 网络框架

原型网络算法采用的神经网络框架是 Vinyals<sup>[2]</sup>等人构造的。网络框架由四个卷积块组成,每个卷积块包括含有 64 个  $3 \times 3$  卷积核<sup>[10]</sup>的卷积层、批归一化层、ReLU 非线性和  $2 \times 2$  最大池化层,如图 2 所示。

### 1.3 原型的计算

支持集经过网络框架<sup>[11]</sup>后变为  $f(S)$ ,其中  $f$  表示神经网络框架,且  $f: x \rightarrow f(x)$ ,  $f(x) \in R^D$ ,然后将每个类的均值作为各自的原型。

表达形式如下:

$$c_k = \frac{1}{m} \sum_{i=1}^m f(x_i), f(x_i) \in f(S_k) \quad (1)$$

其中,  $f(S_k)$  表示支持集中第  $k$  类样本经过网络框架的集合,  $c_k$  表示第  $k$  类的原型。

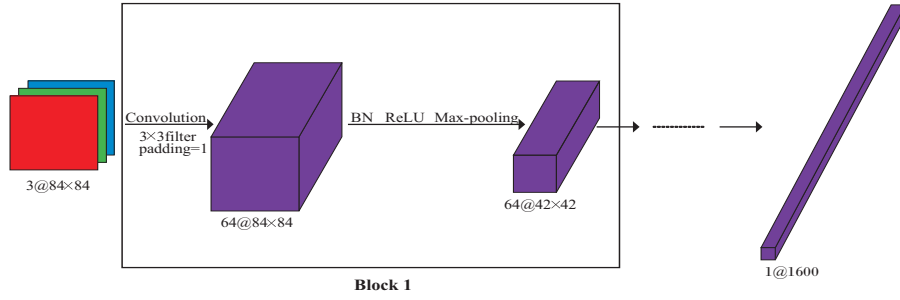


图 2 网络框架

#### 1.4 损失函数的计算

查询集经过网络框架后变为  $f(Q)$ , 分别计算查询点  $f(x_j)$  到每个原型的距离, 利用 softmax 生成它到每个类别的概率, 那么  $f(x_j)$  属于第  $k$  类的概率可表示为:

$$p(y = k | f(x_j)) = \frac{\exp(-d(f(x_j), c_k))}{\sum_k \exp(-d(f(x_j), c_k))} \quad (2)$$

为了最大化  $p(y = k | f(x_j))$ , 需要最小化损失函数  $J$ , 损失函数表示如下:

$$J = -\log(p(y = k | f(x_j))) \quad (3)$$

## 2 密度加权算法

原型网络平等地看待每个类别中的样本, 而不考虑样本本身的特性。然而, 每个类别中不同的样本在代表性上有差别。为此, 该文提出了一种基于密度加权的策略, 每个类别中密度较大的样本, 它们的代表性越大, 赋予它们较大的权重; 密度较小的样本它们的代表性越小, 赋予它们较小的权重。然后根据每个样本赋予的权重, 加权得到原型。该文将利用密度加权算法得到的原型网络称为密度加权原型网络。

在图 1 中以每个样本为中心在临界距离范围<sup>[12]</sup>内的样本数不同, 如果以样本为中心在临界距离范围内的样本数越大, 那么该样本的密度越大, 该样本的代表性越强, 对应图中的以样本为中心在固定距离范围颜色越深。如果以样本为中心在临界距离范围内的样本数越小, 那么该样本的密度越小, 该样本的代表性越弱, 对应图中的以样本为中心在固定距离范围颜色越浅。每个类别利用密度加权算法计算原型的具体流程如下。

#### 2.1 计算距离矩阵

为了度量每个类别的样本的密度<sup>[13]</sup>, 利用距离矩阵来度量各个样本到其他样本的距离, 再对距离矩阵进一步处理得到各个样本距离它们相对较近的样本的个数, 将得到的每个样本相对较近的样本个数进行归一化, 从而得到要求得的权值。距离矩阵具体的数学

表达形式如下:

$$D_k = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1m} \\ d_{21} & d_{22} & \cdots & d_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ d_{m1} & d_{m2} & \cdots & d_{mm} \end{bmatrix} \quad (4)$$

$D_k$  表示第  $k$  类样本的距离矩阵, 其中  $d_{ij} = \|f(x_i) - f(x_j)\|_2$ , 易知距离矩阵是一个对角矩阵, 且主对角线上的元素全为零。

#### 2.2 计算临界距离

为了统计以样本为中心在固定距离范围内的样本数, 需要计算临界距离, 通过距离矩阵<sup>[14]</sup>, 可以得到  $k$  类样本所有的距离组成的集合  $d_k = \{d_{12}, \dots, d_{1m}, d_{23}, \dots, d_{2m}, \dots, d_{m-1,m}\}$ , 该文计算临界距离  $d_k^{\lim}$  的方式如下:

$$d_k^{\lim} = \text{quantile}(d_k) \quad (5)$$

其中,  $\text{quantile}(d_k)$  表示  $d_k$  的分位数。

#### 2.3 计算权重

定义 0-1 函数  $I(x) = \begin{cases} 1 & x \leq d_k^{\lim} \\ 0 & x > d_k^{\lim} \end{cases}$ , 对距离矩阵

$D_k$  的每个元素  $d_{ij}$  分别计算  $I(d_{ij})$ , 然后距离矩阵  $D_k$  中元素全部赋值为 0 或 1, 其中 0 表示元素不在临界距离内, 1 表示元素在临界距离内。对距离矩阵  $D_k$  的每行元素分布求和, 即可得到以每个样本为中心在固定距离范围内的样本数, 然后再对它进行归一化得到权重。具体计算如下:

$$I(D_k) = \begin{bmatrix} I(d_{11}) & I(d_{12}) & \cdots & I(d_{1m}) \\ I(d_{21}) & I(d_{22}) & \cdots & I(d_{2m}) \\ \vdots & \vdots & \ddots & \vdots \\ I(d_{m1}) & I(d_{m2}) & \cdots & I(d_{mm}) \end{bmatrix} \rightarrow \begin{bmatrix} s_{I(d_1)} \\ s_{I(d_2)} \\ \vdots \\ s_{I(d_n)} \end{bmatrix} \rightarrow \begin{bmatrix} w_{k1} \\ w_{k2} \\ \vdots \\ w_{km} \end{bmatrix} \quad (6)$$

令  $w_k = (w_{k1}, \dots, w_{km})'$ , 表示 episode 第  $k$  类中每个样本所对应的权, 其中  $s_{I(d_i)} = \text{sum}(I(d_{i1}), I(d_{i2}), \dots,$

$I(d_{in}))$ ,  $w_{ki} = \frac{s_{I(d_i)}}{\text{sum}(s_{I(d_1)}, s_{I(d_2)}, \dots, s_{I(d_n)})}$ , 当每类样本中存在极少数离群点时, 那么它对应的权值就相对较小, 对计算原型的影响变小。

## 2.4 计算加权原型

利用式(6)第  $k$  类样本的权重  $w_k = (w_{k1}, \dots, w_{km})'$  与第  $k$  类样本进行加权来重新计算原型, 表达形式如下:

$$c_k' = \sum_{i=1}^m w_{ki} f(x_i), \quad f(x_i) \in f(S_k) \quad (7)$$

## 3 密度加权原型网络算法流程

利用密度加权算法来计算原型分为以下几个步骤, 支持集经过网络框架后计算每个类别中各自样本到其他样本之间的距离得到距离矩阵, 利用距离集合来构造临界距离, 利用 0-1 函数作用在距离矩阵的每个元素当中, 对处理后距离矩阵进行行求和, 得到在每个样本在临界距离内的样本数, 进行归一化得到相应的权值, 利用权值与样本进行加权计算原型。利用计算好的原型和经过网络框架的查询集样本带入到公式(2)和(3)中得到损失函数。具体算法流程如下:

算法 1: 密度加权原型网络。

输入: 训练集  $D_{\text{train}} = \{(x_1, y_1), \dots, (x_M, y_M) \mid x_i \in R^{H \times W \times C}, y_i \in \{1, \dots, K\}\}$

输出: 训练损失

1. 生成支持集  $S$ : 从训练集中随机抽取  $N$  个不同的类别, 再从每个类别中抽取  $m$  个不同的样本, 且  $N \leq K$ ;
2. 生成查询集  $Q$ : 从训练集中随机抽取  $N$  个不同的类别, 再从每个类别中抽取  $n$  个不同的样本, 且  $S \cap Q = \emptyset$ ;
3. 支持集和查询集样本经过神经网络框架嵌入到特征空间分别表示为:  $f(S)$ ,  $f(Q)$ ;
4. for  $k$  in  $\{1, \dots, N\}$  do
5. 第  $k$  个类别中样本距离矩阵  $D_k = (d_{ij})_{m \times m}$ ;
6. 通过公式(5)计算  $d_k^{\text{lim}}$ ;
7. 由公式(6)定义  $I(x)$ , 并通过计算公式(6)计算  $w_k$ ;
8. 由公式(7)计算  $c_k'$ ;
9. end for
10.  $J \leftarrow 0$
11. for  $k$  in  $\{1, \dots, N\}$  do
12. for  $(f(x_j), y_j)$  in  $f(Q_k)$  do
13. 通过公式(2)计算  $p(y = k \mid f(x_j))$ ;
14. 通过公式(3)计算  $J$ ;
15. end for
16. end for

## 4 实验及结果分析

实验主要对训练和测试过程中得到的结果进行分析。在训练期间保存好每次训练的模型并在验证集上

得到相应损失值和准确率, 将它们可视化进行分析。在测试期间建立评估指标, 分别为准确率的置信区间和平均准确率, 准确率的置信区间表示对测试期间 episodes 得到的准确率的均值的区间估计, 平均准确率表示计算前  $n$  次迭代的平均准确率。将测试集带入到训练期间得到的最大准确率的模型中, 计算每次迭代得到的平均准确率和所有迭代得到的准确率的 95% 置信区间, 对比模型之间的差异。

### 4.1 实验说明

miniImageNet 是来自 ImageNet 的一个子集, 由 60 000 张大小为  $84 \times 84$  的 RGB 图像组成, 分为 100 个类别, 每个类别有 600 个样本。将整个数据集划分为训练集、验证集和测试集, 采用 Ravi 和 Larochelle<sup>[2]</sup> 分割的方法, 他们将 100 个类别进行划分, 64 个类别用于训练, 16 个类别用于验证, 20 个类别用于测试。将每张  $84 \times 84$  的 RGB 图片带入原型网络框架<sup>[15]</sup> 中会输出一张  $1 \times 1 \times 600$  的向量, 如图 2 所示。训练网络框架利用随机梯度下降法通过 Adam 进行优化, 将初始学习率设置为 0.001, 每 2 000 个 episodes 将学习率降低一半。并且仅使用批量归一化, 没有使用正则化。在训练和验证期间设置了 5-way 5-shot 和 20-way 5-shot, 对于 5-way 5-shot 实验在 kaggle 网站内置的 GPU 环境, NVIDIA 发布的 PCI-E Tesla P100, 网络框架在 Pytorch 1.6.0 进行训练, 对于 20-way 5-shot 实验, 配置为 RTX2080, 网络框架在 Pytorch 1.6.0 进行训练

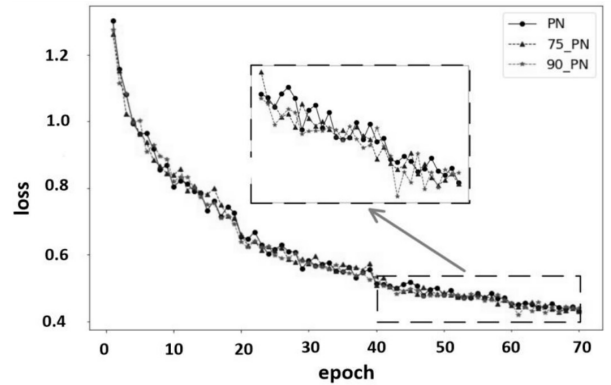


图3 5-way 5-shot 分类损失曲线

### 4.2 训练过程结果分析

在训练和验证期间设置了 5-way 5-shot 和 20-way 5-shot, 每个类别的查询点的个数为支持点个数的 3 倍, 总共 600 个 episodes。测试期间的 episodes 和训练时期的一致。临界距离的不同会对加权原型的计算产生很大的影响, 经过反复实验比较 70% ~ 95% 分位数效果较好, 该文选取了 75%、90% 分位数计算临界距离, 得到密度加权原型网络<sup>[16]</sup>, 结果如图 3 和图 4 所示, 图中 PN, 75\_PN 和 90\_PN 分别表示原型网络算



法,90%分位数和75%分位数计算得到的密度加权原型网络算法。可以看出利用90%分位数和75%分位数计算临界距离得到的密度加权网络的损失函数随着迭代次数的增加,总体位于原型网络的损失函数的下方;其准确率随着迭代次数的增加,总体位于原型网络的准确率的上方,所以密度加权原型网络相对于原型网络更优。

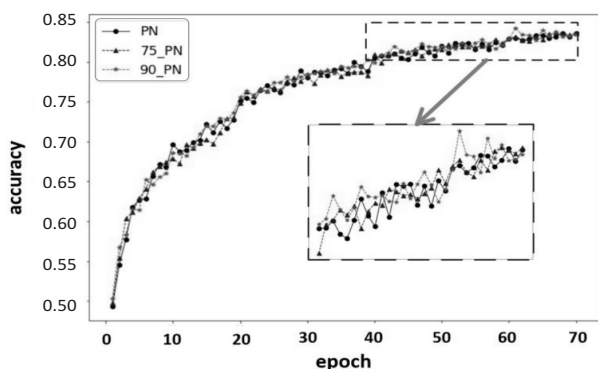


图 4 5-way 5-shot 分类准确率曲线

#### 4.3 测试过程结果分析

为了进一步与原型网络进行比较,对于5-way 5-shot分类,将测试集带入到训练好的模型进行测试。测试阶段采用的 episodes 和训练阶段的一样,进行150次迭代,当进行第 $n$ 次迭代时,计算前 $n$ 次迭代的平均准确率,结果如图5所示。可见利用90%分位数和75%分位数计算临界距离得到的密度加权原型网络测试的平均准确率比利用简单的均值来计算原型得到原型网络的平均准确率高;随着迭代次数的增加,密度加权原型网络测试平均准确率明显优于原型网络,当迭代次数大于25时这种趋势比较明显。

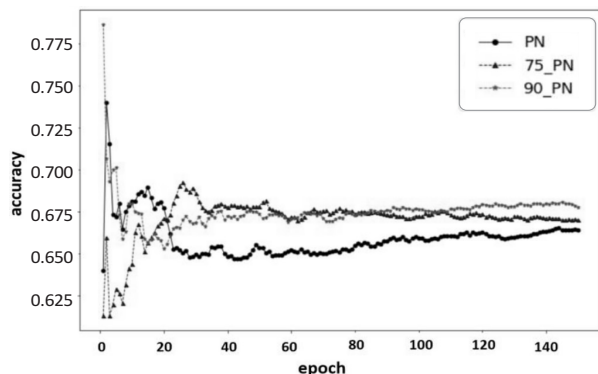


图 5 5-way 5-shot 分类平均准确率曲线

对于20-way 5-shot分类,随着样本量的提升,准确率也有很大幅度的提升。将测试集带入到训练好的模型进行测试。测试阶段采用的 episodes 和训练阶段的一样,进行200次迭代,当进行第 $n$ 次迭代时,计算前 $n$ 次迭代的平均准确率,结果如图6所示。可见利用90%分位数和75%分位数计算临界距离得到的密度加权原型网络的测试平均准确率比利用简单的均值

来计算原型得到原型网络的平均准确率高;随着迭代次数的增加,密度加权原型网络测试平均准确率明显优于原型网络,当迭代次数大于50时这种趋势比较明显。

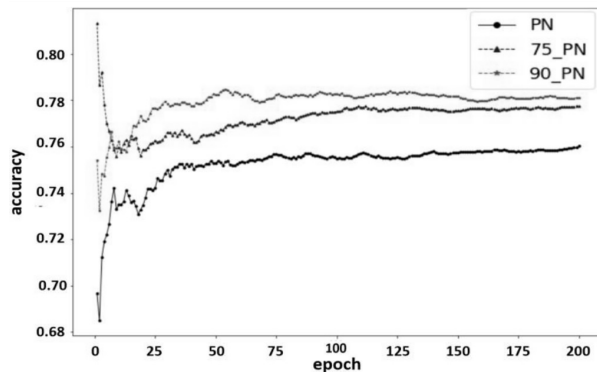


图 6 20-way 5-shot 分类平均准确率曲线

将5-way 5-shot,20-way 5-shot分类实验设置其他的模型性能的评估指标,与原型网络算法进行对比,从测试集中随机生成的200个 episodes 进行测试,计算每次 episode 的准确率,并得到200个 episodes 准确率的95%置信区间,具体结果见表1。由于实验条件和其他因素不一样,该文原型网络的结果与原型网络作者得到的结果存在着一些差异。如表1所示,密度加权原型网络优于原型网络,且利用90%分位数得到的密度加权原型网络比原型网络大约提升1个百分点。

表 1 miniImageNet 上分类精度

小样本 学习算法	准确率/%	
	5-way 5-shot	20-way 5-shot
PN	66.51±1.22	77.17±0.59
75_PN	67.24±1.19	77.73±0.62
90_PN	67.51±1.16	78.11±0.53

## 5 结束语

为了减小离群点对原型计算的影响,并在计算原型的过程中充分利用每个样本的特性,提出了原型网络算法的改进方法,在计算原型的方法中采用密度加权算法来计算原型。密度加权算法的思想是对密度大的样本赋予较大的权值,对密度小的样本赋予较小的权值。而对于密度的衡量关键取决于对临界距离的选取。由于不同临界距离内的样本数是不同的,距离越大,每个样本所包含的样本数越多,那么对于一些偏离总体的样本对应的密度相对较小,从而偏离总体的样本对原型的计算影响较小;距离越小每个样本所包含的样本数较少,使得密度的区分相对不明显,原型的计算变化不明显当取分位数最大最小时每个样本的密度

一样,从而与利用均值计算原型一致。原型网络算法的核心在于优化网络框架,使得样本经过网络框架后类内的差异更小,类间的差异更大,更容易分类,在今后的工作中为了进一步提升密度加权网络算法的精度,将对损失函数的进行改进来达到这个目的。

#### 参考文献:

- [1] WANG Yaqing, YAO Quanming, KWOK J T, et al. Generalizing from a few examples; a survey on few-shot learning [J]. ACM Computing Surveys, 2020, 63(3): 15–20.
- [2] CUBUK E D, ZOPH B, MANE D, et al. AutoAugment: learning augmentation policies from data [C]//Proceedings of the conference on computer vision and pattern recognition. Besky: [s. n.], 2019: 113–123.
- [3] WEI J, ZOU K. EDA: easy data augmentation techniques for boosting performance on text classification tasks [C]//Proceedings of the conference on empirical methods in natural language processing and international joint conference on natural language processing. Aice: [s. n.], 2019: 6383–6389.
- [4] VINIYALS O, BLUNDELL C, LILLICRAP T, et al. Matching networks for one shot learning [C]//Advances in neural information processing systems. [s. l.]: [s. n.], 2016: 3637–3645.
- [5] SNELL J, SWERSKY K, ZEMEL R S. Prototypical networks for few-shot learning [C]//Proceedings of the 31rd conference on neural information processing systems. Seoul, Korea (South): [s. n.], 2017: 4080–4090.
- [6] SUNG F, YANG Y, ZHANG L, et al. Learning to compare: relation network for few-shot learning [C]//Proceedings of the conference on computer vision and pattern recognition. Salt Lake: IEEE, 2018: 1199–1208.
- [7] FINN C, ABBEEL P, LEVINE S. Model-agnostic meta-learning for fast adaptation of deep networks [C]//Proceedings of international conference on machine learning (ICML). Chinesy: [s. n.], 2017: 1126–1135.
- [8] FINN C, XU K, LEVINE S. Probabilistic model-agnostic meta-learning [C]//Advances in neural information processing systems. [s. l.]: MIT Press, 2018: 9537–9548.
- [9] BOTTOU L, CURTIS F E, NOCEDAL J. Optimization methods for large-scale machine learning [J]. SIAM Review, 2018, 60(2): 223–311.
- [10] YATZIV L, BARTESAGHI A, SAPIRO G. Implementation of the fast marching algorithm [J]. Journal of Computational Physics, 2005, 212(2): 393–399.
- [11] ZHAO M, KANG M, TANG B, et al. Deep residual networks with dynamically weighted wavelet coefficients for fault diagnosis of planetary gearboxes [J]. IEEE Transactions on Industrial Electronics, 2017, 65(5): 4290–4300.
- [12] CHEN H, CHEN P, CHEN W, et al. Wind turbine gearbox fault diagnosis based on improved EEMD and Hilbert square demodulation [J]. Applied Sciences, 2017, 7(2): 128–134.
- [13] 贾瑞玉, 宋建林. 基于聚类中心优化的 K-means 最佳聚类数确定方法 [J]. 微电子学与计算机, 2016, 33(5): 62–66.
- [14] 左进, 陈泽茂. 基于改进 K 均值聚类的异常检测算法 [J]. 计算机科学, 2016, 43(8): 258–261.
- [15] RAZAVIAN A S, AZIZPOUR H, SULLIVAN J, et al. CNN features off-the-shelf: an astounding baseline for recognition [C]//IEEE conference on computer vision and pattern recognition. San Francisco: IEEE, 2014: 806–813.
- [16] 刘同明. 数据挖掘技术及其应用 [M]. 北京: 国防工业出版社, 2001: 45–67.