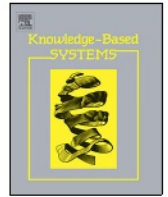




Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



具有残差结构的模糊粒度深度卷积网络

何林杰, 陈玉民*, 吴克寿

厦门理工大学计算机科学与技术学院, 厦门, 361024

条信息

文章历史:

2022年9月4 2022年9月25日接受, 2022年10月3日在线发布

关键词: 模糊集颗粒计算模糊颗粒卷积模糊网络残差结构

摘要

近年来, 深度神经网络技术发展迅速, 在处理和图像、视频、声音等诸多方面都取得了有效的效果。然而, 传统的神经网络只学习样本本身, 忽略了样本之间的差异。当网络深度过深时, 可能会出现网络退化现象。为了解决这些弊端, 我们提出了具有残差结构的模糊粒度深度卷积网络。首先, 我们定义了模糊粒的概念、操作和关联度量, 并通过随机采样构建了模糊粒化的粒化参考系统; 进而定义了粒神经元、粒激活函数、粒卷积、粒残差等模块, 构建了具有残差结构的模糊粒深度卷积网络; 针对粒神经网络设计了损失函数和学习算法, 成功训练了模糊粒神经网络。模糊粒度神经网络具有多粒度、多角度、结构化的特点, 具有良好的泛化性能。最后, 在Cifar100和Tiny Imagenet数据集上进行实验。实验结果表明了残差粒度神经网络的有效性。我们进一步发现, 粒神经网络可以缓解退化问题。而且, 隐藏层更少的粒度神经网络可以实现隐藏层更多的神经网络的功能。

©2022 Elsevier B.V. 版权所有。

1. 介绍

模糊集和模糊计算的概念是由美国科学家Zadeh[1]提出的, 其中模糊集通过隶属函数表示元素与集合之间的关系。模糊集合论是经典集合论的扩展。1979年, Zadeh首次引入了颗粒化和颗粒计算的概念。加拿大院士Pedrycz提出了多种粒度分类[2]和粒度聚类算法[3]。1985年Hoboss[4]发表了一篇题为“粒度”的文章, 用不同的粒度来描述真实的现象, 解决复杂的问题。1997年Zadeh[5]指出模糊逻辑和人类推理逻辑类似于人类推理逻辑。模糊信息粒化是基于人类推理的方法对信息进行粒化。除了模糊信息粒化, 还有很多其他的粒化方法, 比如粗糙集粒化和阴影集粒化。Pedrycz对这些不同条件下的粒化方法[6]进行了总结和讨论。Lin成功地将粒计算应用于知识发现[7]和数据挖掘[8]。Yao通过定义邻域关系来构建邻域粒化[9,10]。2002年, Miao提出了一种基于知识的粒化方法[11], 可以

有效处理属性近似问题[12]。Yager[13]指出, 粒计算的思想对推动智能系统的发展具有重要意义。Qian等人提出了多粒度融合学习[14]来学习不同粒度模式下的决策。2014年, Qian[15]等人设计了一种并行化属性逼近算法, 提高了属性逼近问题的粒度计算运算效率。Sanchez等[16]基于不确定信息理论提出了区间2型模糊信息颗粒的造粒方法。Patricia Melin[17]提出了一种基于多目标分层遗传算法的模块化神经网络优化新方法。Melin将其应用于人脸检测、耳朵识别和其他模式识别任务。陈玉民从集合和向量的角度定义了颗粒的结构, 并进一步研究了颗粒的不确定性和距离度量[18]。2020年, Daniela Sanchez等人提出了基于粒子群优化变量和模糊动态参数自适应的模块化颗粒神经网络[19], 并将其应用于人类识别任务。Li[20]将提升算法与颗粒KNN算法相结合, 进一步提高了KNN算法的性能。颗粒计算是从人类认知的角度定义的一种算法模型, 与人类的逻辑、认知和记忆高度相似, 已成功应用于多个领域[21-27]。

*Corresponding author.

E-mail address: cym0620@163.com (Y. Chen).

1943年, Mcculloch和Pitts通过分析模拟生物的神经网络特征, 首次提出了人工神经网络的概念。1958年, Rosenbaltt提出了感知器模型[28], 实现了单个神经元。感知器模型的提出, 将神经网络的发展带入了第一个高潮。神经网络的第二个高潮是伴随着80年代的连接主义。通过使用反向传播算法, Rumelhart使得成功训练具有一到两层隐藏层[29]的神经网络成为可能。Hochreiter等人提出了LSTM结构, 使神经网络应用于时间序列数据建模[30]成为可能。LeCun提出了LeNet-5网络, 以及后来生成的变体, 定义了现代卷积神经网络的基本结构[31]。2006年, Hinton提出了深度信念网络[32], 通过贪婪的逐层训练解决了深度神经网络难以训练的问题。深度信念网络的提出, 重新点燃了神经网络的浪潮。2012年, AlexNet[33]网络在ImageNet图像识别竞赛中以远低于第二名的成绩获胜, 引起了学术界和工业界的关注。Kai-Ming He等人提出了残差网络[34], 在某种程度上解决了神经网络过深导致梯度消失和网络退化的问题。残差网络一经提出, 就受到了相关研究者的广泛关注。残差网络已经被应用到多个领域[35-39], 并且出现了许多变体[40,41]。尽管如此, 传统的神经网络只从当前的样本中学习。数据集中多个分离率样本的特征之间仍然存在一些联系, 而传统神经网络忽略了这些联系。当网络深度过深时, 可能会出现网络退化现象。为了解决这些弱点, 我们设计了一种带有残差结构的模糊粒度深度卷积网络。粒深度卷积网络的多角度学习模式使神经网络具有更好的泛化性能, 可以用更少的粒隐藏层实现更多的隐层神经网络功能。本文采用的粒化方法属于模糊粒化, 但与一般的模糊粒化方法有所不同。该算法的模糊粒化是基于随机选取的参考样本。粒化模式是部分粒化而不是全局粒化。为了构建粒神经网络, 一些粒提出的操作也不同于传统的模糊信息粒操作。基于粒的结构性质, 粒神经元自然具有可并行性。而且, 采用随机参考粒化, 节省了粒化时间和空间。

本文的贡献如下。首先, 我们定义了一种基于特征相似性的粒化方法。然后我们还定义了各种粒度操作和粒度度量。在此之后, 提出了粒神经元和粒激活函数。以同样的方式, 定义了粒残差结构和粒卷积。进一步, 我们提出了粒残差网络及其优化算法。最后, 在实验中证明了粒残差网络的有效性。

本文第1节简要介绍了粒计算和神经网络的背景。在第2节中, 我们描述了粒、粒操作和粒度度量的定义。然后, 第3节设计了粒神经元、粒激活函数、粒卷积和粒残差块的结构, 以构建具有残差结构的模糊粒深度卷积网络。此外, 第4节在多个数据集上展示了所提算法的有效性。最后一部分是对整篇论文的总结。

2. 模糊粒的定义、操作和度量

2.1. 模糊造粒

模糊集是处理不确定性信息的有效工具。对于信息系统 $U = (X, C, D)$, $X = \{X_1, X_2, \dots, x_n\}$ 为样本集, $C = \{C_1, C_2, \dots, c_m\}$ 是样本对应的特征集, D 是样本对应的决策集。有的样本 $P = \{P_1, P_2, \dots, p_k\} \subseteq X$ 从 X 中随机选取 X 作为参考样本集。对于给定的样本 $x \in X$, 其中单特征 $c \in C$, $v(x, c) \in [0, 1]$ 表示在特征 c 上归一化的样本 x 的值。

定义1. 样本 $X \in X$, $p \in P$, 单特征 $C \in C$ 的信息系统 $U = (X, C, D)$ 。 x 与 p 在特征 c 上的相似度为:

$$S_c(x, p) = 1 - |v(x, c) - v(p, c)|. \quad (1)$$

定义2. 对于信息系统 $U = (X, C, D)$, 我们对样本 $X \in X$ 进行模糊粒化, 参考样本集 $P = \{P_1, P_2, \dots, p_k\}$ 对任意特征 $c \in C$ 将样本 x 和参考样本集 P 在特征 c 上进行粒化, 形成模糊粒。 x 在 c 上的模糊粒定义为:

$$g_c(x) = \{g_c(x)_j\}_{j=1}^k = \{r_j\}_{j=1}^k = \{r_1, r_2, \dots, r_k\}, \quad (2)$$

$$\text{where } r_j = S_c(x, p_j) = 1 - |v(x, c) - v(p_j, c)|. \quad (3)$$

$g_c(x)$ 是一个集合。 r_j 是集合中的一个元素。而 r_j 是样本 x 与参考样本 p_j 在特征 c 上的相似度, 从定义1很容易知道 $r_j = S_c(x, p_j) \in [0, 1]$ 。我们定义 $g_c(x)$ 为模糊粒, $g_c(x)_j$ 为模糊粒的第 j 个核 $g_c(x)$, 粒由粒核组成。如果 $\{r_j\}_{j=1}^k = \{0, 0, 0, \dots, 0\}$, 我们将 $\{r_j\}_{j=1}^k$ 声明为 0-模糊粒。而如果 $\{r_j\}_{j=1}^k = \{1, 1, 1, \dots, 1\}$, 我们将 $\{r_j\}_{j=1}^k$ 声明为 1-模糊粒。

定义3. 对于任意 $X \subseteq X$ 和参考样本集 P 的信息系统 $U = (X, C, D)$, 任意特征子集 $B = \{B_1, B_2, \dots, b_m\} \subseteq X$, X 在特征子集 B 上的模糊粒向量为:

$$G(x) = (g_{b1}(x), g_{b2}(x), \dots, g_{bm}(x))^T. \quad (4)$$

$g_{bm}(x)$ 是 x 在特征上的模糊粒 b_m , 缩写为 $G(x) = (G_1(x), G_2(x), \dots, g_m(x))^T$ 。

模糊粒向量由模糊粒组成, 模糊粒由粒核组成。所以模糊粒向量也可以用粒核矩阵来表示。

$$G(x) = \begin{bmatrix} g_1(x)_1 & g_1(x)_2 & \dots & g_1(x)_k \\ g_2(x)_1 & g_2(x)_2 & \dots & g_2(x)_k \\ \dots & \dots & \dots & \dots \\ g_m(x)_1 & g_m(x)_2 & \dots & g_m(x)_k \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1k} \\ r_{21} & r_{22} & \dots & r_{2k} \\ \dots & \dots & \dots & \dots \\ r_{m1} & r_{m2} & \dots & r_{mk} \end{bmatrix} \quad (5)$$

模糊粒向量 $G(x)$ 是模糊粒的集合, 而常规向量的元素是实数。

例1. 信息系统 $U = (X, C, D)$ 如表1所示, 其中 $X = \{X_1, X_2, X_3, X_4\}$, $C = \{a, b, c\}$, $D = \{\text{狗}, \text{猫}\}$ 。

表1

信息系统的一个例子。

U	a	b	c	dog	cat
x_1	0.3	0.6	0.6	1	0
x_2	0.5	0.4	0.5	1	0
x_3	0.1	0.2	0.9	0	1
x_4	0.4	0.7	0.1	0	1

设参考集为 $P = \{x_1, x_3\}$ 。样本集 $X = \{X_1, X_2, X_3, x_4\}$ 对特征 a 的模糊粒化是:

$$g_a(x_1) = \{1, 0.8\}, g_a(x_2) = \{0.8, 0.6\}, g_a(x_3) = \{0.8, 1\}, g_a(x_4) = \{0.9, 0.7\}. \quad (6)$$

样本集 $X = \{X_1, X_2, X_3, x_4\}$ 对特征 b 的模糊粒化结果为:

$$g_b(x_1) = \{1, 0.6\}, g_b(x_2) = \{0.8, 0.8\}, g_b(x_3) = \{0.6, 1\}, g_b(x_4) = \{0.9, 0.5\}. \quad (7)$$

样本集 $X = \{X_1, X_2, X_3, x_4\}$ 在特征 c 上的模糊粒化结果为:

$$g_c(x_1) = \{1, 0.7\}, g_c(x_2) = \{0.9, 0.6\}, g_c(x_3) = \{0.7, 1\}, g_c(x_4) = \{0.5, 0.2\}. \quad (8)$$

样本 $X = (X_1, X_2, X_3, x_4)$ 的粒化向量为:

$$G_{[a,b,c]}(x_1) = (g_a(x_1), g_b(x_1), g_c(x_1))^T = (\{1, 0.8\}, \{1, 0.6\}, \{1, 0.7\})^T, \quad (9)$$

$$G_{[a,b,c]}(x_2) = (g_a(x_2), g_b(x_2), g_c(x_2))^T = (\{0.8, 0.6\}, \{0.8, 0.8\}, \{0.9, 0.6\})^T, \quad (10)$$

$$G_{[a,b,c]}(x_3) = (g_a(x_3), g_b(x_3), g_c(x_3))^T = (\{0.8, 1\}, \{0.6, 1\}, \{0.7, 1\})^T, \quad (11)$$

$$G_{[a,b,c]}(x_4) = (g_a(x_4), g_b(x_4), g_c(x_4))^T = (\{0.9, 0.7\}, \{0.9, 0.5\}, \{0.5, 0.2\})^T. \quad (12)$$

2.2. 模糊粒度操作和指标

本小节给出了粒的操作和度量。

定义4. 设 $g_a(x)$ 、 $g_b(x)$ 分别为样本 x 对特征 a 、 b 的两个模糊粒，则模糊粒的加、减、乘、除运算如下:

$$g_a(x) + g_b(x) = \{g_a(x)_j + g_b(x)_j\}_{j=1}^k = \{g_a(x)_1 + g_b(x)_1, g_a(x)_2 + g_b(x)_2, \dots, g_a(x)_k + g_b(x)_k\}, \quad (13)$$

$$g_a(x) - g_b(x) = \{g_a(x)_j - g_b(x)_j\}_{j=1}^k = \{g_a(x)_1 - g_b(x)_1, g_a(x)_2 - g_b(x)_2, \dots, g_a(x)_k - g_b(x)_k\}, \quad (14)$$

$$g_a(x) * g_b(x) = \{g_a(x)_j * g_b(x)_j\}_{j=1}^k = \{g_a(x)_1 * g_b(x)_1, g_a(x)_2 * g_b(x)_2, \dots, g_a(x)_k * g_b(x)_k\}, \quad (15)$$

$$g_a(x) / g_b(x) = \{g_a(x)_j / g_b(x)_j\}_{j=1}^k = \{g_a(x)_1 / g_b(x)_1, g_a(x)_2 / g_b(x)_2, \dots, g_a(x)_k / g_b(x)_k\}. \quad (16)$$

例2. 信息系统 $U = (X, C, D)$ 如表1所示。从例1中，很容易知道:

$$g_a(x_2) = \{0.8, 0.6\}. \quad (17)$$

$$g_b(x_2) = \{0.8, 0.8\}. \quad (18)$$

According to Definition 4:

$$g_a(x_1) + g_b(x_1) = \{0.8 + 0.8, 0.6 + 0.8\} = \{1.6, 1.4\}. \quad (19)$$

$$g_a(x_1) - g_b(x_1) = \{0.8 - 0.8, 0.6 - 0.8\} = \{0, -0.2\}. \quad (20)$$

$$g_a(x_1) * g_b(x_1) = \{0.8 * 0.8, 0.6 * 0.8\} = \{0.64, 0.48\}. \quad (21)$$

$$g_a(x_1) / g_b(x_1) = \{0.8/0.8, 0.6/0.8\} = \{1, 0.75\}. \quad (22)$$

定义5. 设 $g_a(x)$ 、 $g_a(y)$ 为样本 x 在特征 a 上的两个模糊粒，则模糊粒的加、减、乘、除运算如下:

$$g_a(x) + g_a(y) = \{g_a(x)_j + g_a(y)_j\}_{j=1}^k = \{g_a(x)_1 + g_a(y)_1, g_a(x)_2 + g_a(y)_2, \dots, g_a(x)_k + g_a(y)_k\}, \quad (23)$$

$$g_a(x) - g_a(y) = \{g_a(x)_j - g_a(y)_j\}_{j=1}^k = \{g_a(x)_j - g_a(y)_j, g_a(x)_j - g_a(y)_j, \dots, g_a(x)_j - g_a(y)_j\}, \quad (24)$$

$$G_a(x) * G_a(y) = \{G_a(x)_j * G_a(y)_j\}_{j=1}^k = \{g_a(x)_1 * g_a(y)_1, g_a(x)_2 * g_a(y)_2, \dots, g_a(x)_k * g_a(y)_k\}, \quad (25)$$

$$g_a(x) / g_a(y) = \{g_a(x)_j / g_a(y)_j\}_{j=1}^k = \{g_a(x)_1 / g_a(y)_1, g_a(x)_2 / g_a(y)_2, \dots, g_a(x)_k / g_a(y)_k\}. \quad (26)$$

例3. 信息系统 $U = (X, C, D)$ 如表1所示。从例1中，很容易知道:

$$g_a(x_1) = \{1, 0.8\}. \quad (27)$$

$$g_a(x_2) = \{0.8, 0.6\}. \quad (28)$$

According to Definition 5:

$$g_a(x_1) + g_a(x_2) = \{1 + 0.8, 0.8 + 0.6\} = \{1.8, 1.4\}. \quad (29)$$

$$g_a(x_1) - g_a(x_2) = \{1 - 0.8, 0.8 - 0.6\} = \{0.2, 0.2\}. \quad (30)$$

$$g_a(x_1) * g_a(x_2) = \{1 * 0.8, 0.8 * 0.6\} = \{0.8, 0.48\}. \quad (31)$$

$$g_a(x_1) / g_a(x_2) = \{1/0.8, 0.8/0.6\} = \{1.25, 1.3334\}. \quad (32)$$

两个模糊粒第四次运算的结果也是一个模糊粒。定义4中的4个基本运算是同一样本的模糊粒在不同特征上的运算。定义5中的运算是不同样本在同一特征上的模糊粒上的运算。

定义6. 设模糊粒向量为 $G(x) = (g_1(x), g_2(x), \dots, g_m(x))^T$ 和 $G(y) = (g_1(y), g_2(y), \dots, g_m(y))^T$ ，那么两个模糊粒向量的点积定义为:

$$G(x) \cdot G(y) = g_1(x) * g_1(y) + g_2(x) * g_2(y) + \dots + g_m(x) * g_m(y). \quad (33)$$

例4. 假设:

$$G(x) = (g_1(x), g_2(x), g_3(x))^T, G(y) = (g_1(y), g_2(y), g_3(y))^T. \quad (34)$$

$$g_1(x) = \{1, 1, 1\}, g_2(x) = \{1, 2, 3\}, g_3(x) = \{0, 1, 0\}. \quad (35)$$

$$g_1(y) = \{2, 2, 2\}, g_2(y) = \{0, 4, 2\}, g_3(y) = \{2, 4, 6\}. \quad (36)$$

从定义6可以看出:

$$G(x) \cdot G(y) = \{1*2, 1*2, 1*2\} + \{1*0, 2*4, 3*2\} + \{0*2, 1*4, 0*6\} \quad (37)$$

$$G(x) \cdot G(y) = \{2, 2, 2\} + \{0, 8, 6\} + \{0, 4, 0\} \quad (38)$$

$$G(x) \cdot G(y) = \{2, 12, 8\} \quad (39)$$

两个模糊粒向量的点积得到一个模糊粒。设模糊粒向量为 $G(x) = (G_1(x), G_2(x), G_3(x), \dots, g_m(x), 1)$, 权重模糊粒向量为 $W = (W_1, W_2, W_3, \dots, w_m, b)$, 其点积为

$$W \bullet G(x) = w_1 * g_1(x) + w_2 * g_2(x) + \dots + w_m * g_m(x) + 1 * b. \quad (40)$$

定义7. 设模糊粒为 $g(x) = \{r_j\}_{j=1}^k$, 则模糊粒函数定义为:

$$f(g(x)) = f(\{r_j\}_{j=1}^k) = \{f(r_1), f(r_2), \dots, f(r_k)\}. \quad (41)$$

定义8. 设模糊颗粒为 $g(x) = \{r_j\}_{j=1}^k$, 其质量为:

$$Quality(g(x)) = \sum_{j=1}^k r_j. \quad (42)$$

定义9. 设 m 维模糊粒向量为 $G = (G_1, G_2, \dots, g_m)$, 则模糊粒向量范数粒定义为:

(a). 模糊粒向量的1-范数:

$$\|G\|_1 = \sum_{i=1}^m |g_i|. \quad (43)$$

(b). 模糊粒向量的2-范数:

$$\|G\|_2 = \sqrt{\sum_{i=1}^m g_i * g_i} = \sqrt{G \cdot G}. \quad (44)$$

(c). 模糊粒向量的 p -范数:

$$\|G\|_p = (\sum_{i=1}^m g_i^p)^{\frac{1}{p}}. \quad (45)$$

模糊粒向量的范数运算的结果是模糊粒。范数操作提供了一种将模糊粒向量转换为模糊粒的方法。

3. 模糊粒残差网络

3.1. 颗粒激活函数和颗粒神经元

激活函数对于神经网络训练学习复杂的非线性函数很重要, 它为神经网络提供了一个重要的非线性因素。为了使神经网络也能拟合各种复杂的非线性函数, 我们为神经网络引入了几个非线性的粒激活函数。粒激活函数的输入和输出都是模糊粒。

在传统的神经网络中, 常用的激活函数有sigmoid函数、tanh函数、ReLU函数、Leaky ReLU函数等。分别定义了粒状sigmoid函数、粒状tanh函数、粒状ReLU函数和粒状Leaky ReLU函数。

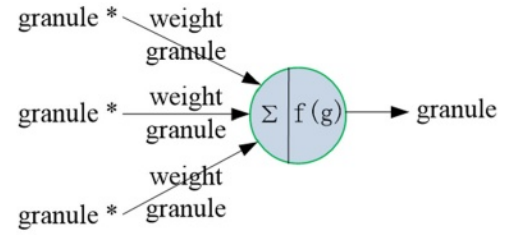


图1所示。颗粒状神经元。

定义10. 设模糊粒为 $g(x) = \{r_j\}_{j=1}^k$ 它的。

颗粒sigmoid函数、颗粒tanh函数、颗粒ReLU函数、颗粒Leaky ReLU函数分别为:

$$f_1(g(x)) = \left\{ \frac{1}{1 + e^{-r_j}} \right\}_{j=1}^k. \quad (46)$$

$$f_2(g(x)) = \left\{ \frac{e^{r_j} - e^{-r_j}}{e^{r_j} + e^{-r_j}} \right\}_{j=1}^k. \quad (47)$$

$$f_3(g(x)) = \left\{ \max(0, r_j) \right\}_{j=1}^k. \quad (48)$$

$$f_4(g(x)) = \left\{ \begin{array}{ll} \alpha r_j, & x < 0 \\ r_j, & x \geq 0 \end{array} \right\}_{j=1}^k. \quad (49)$$

粒状激活函数的导数分别为:

$$f_1'(g(x)) = \left\{ \frac{d}{dr_j} f_1(r_j) \right\}_{j=1}^k = \left\{ \frac{1}{1 + e^{-r_j}} (1 - \frac{1}{1 + e^{-r_j}}) \right\}_{j=1}^k. \quad (50)$$

$$f_2'(g(x)) = \left\{ \frac{d}{dr_j} f_2(r_j) \right\}_{j=1}^k = \left\{ 1 - (\frac{e^{r_j} - e^{-r_j}}{e^{r_j} + e^{-r_j}})^2 \right\}_{j=1}^k. \quad (51)$$

$$f_3'(g(x)) = \left\{ \frac{d}{dr_j} f_3(r_j) \right\}_{j=1}^k = \left\{ \begin{array}{ll} 0, & \alpha r_j < 0 \\ 1, & \alpha r_j = 0 \end{array} \right\}_{j=1}^k. \quad (52)$$

$$f_4'(g(x)) = \left\{ \frac{d}{dr_j} f_4(r_j) \right\}_{j=1}^k = \left\{ \begin{array}{ll} \alpha, & \alpha r_j < 0 \\ 1, & \alpha r_j = 0 \\ \text{undefined}, & \alpha r_j > 0 \end{array} \right\}_{j=1}^k. \quad (53)$$

如图1所示, 颗粒神经元类似于传统神经元。粒神经元的输入是一个模糊粒向量, 输出是模糊粒。将模糊粒向量和权重粒向量点乘, 得到一个模糊粒。再将模糊粒放入粒激活函数中, 得到输出的模糊粒。

3.2. 颗粒卷积

定义11. 设模糊粒向量为 $G = (G_1(x), G_2(x), \dots, g_m(x))$, 核粒向量为 $K = (K_1, K_2, \dots, k_n)$, 粒向量卷积定义为:

$$S(G, K) = (G \bullet K)[m] = \sum_n G[m+n] \cdot K[n]. \quad (54)$$

当我们训练一个粒卷积网络时, G 和 K 执行一个卷积操作, 输出一个如图2所示的粒向量 S 。当误差反向传播时, 会生成一个误差粒度向量 $E = \partial \partial_s L$, 其中 L 是粒度损失函数。为了修正核粒度向量 K 的权值, 我们需要对其权值进行导数运算。

$$\frac{\partial L}{\partial K} = \sum_n E \cdot G[m+n]. \quad (55)$$

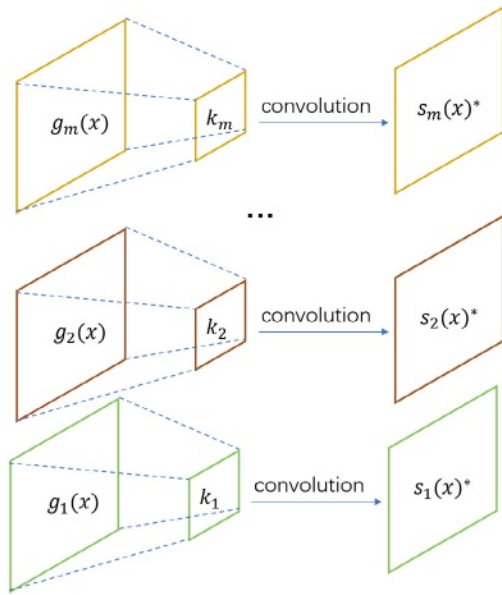
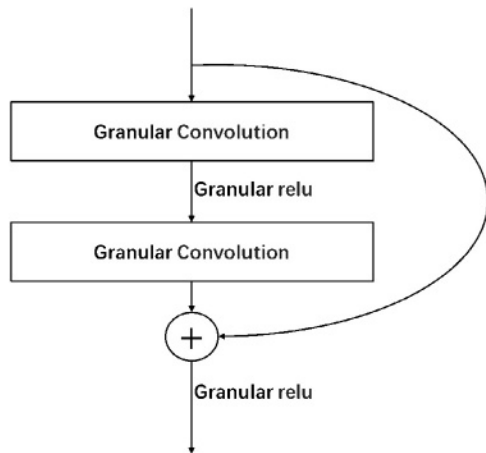


图2所示。颗粒卷积操作。

图3所示。颗粒状残余结构。



3.3. 颗粒状残余结构

为了解决神经网络过深时的退化问题，提出了粒残差块。对于普通的神经网络，输入的模糊粒向量为 $G(x)$ ，经过神经网络学习后的特征粒向量为 $D(G(x))$ 。我们将粒残差定义为 $F(G(x)) = D(G(x)) - G(x)$ ，那么粒残差块学习到的特征粒向量为 $D(G(x)) + G(x)$ 。粒残差块如图3所示。

3.4. 颗粒神经网络损失函数

在深度学习中，设计损失函数或成本函数是一个非常重要的方面。损失函数的作用是评估模型的预测值和真实值之间的差异。对于一般的分类网络来说，神经网络的损失值越小，意味着模型分类效果越好。

在神经网络中，输入是一个特征模糊粒向量，输出是一个决策模糊粒向量。我们将样本标签的onehot编码扩展到带标签的粒向量。决策模糊粒向量可以与softmax函数后的标签模糊粒向量进行比较，构成了粒损失函数。

定义12. 令决策模糊粒向量 $G = \{G_1, G_2, G_3, \dots, g_i\}$ ，其中 $g_i = \{r_j\}_{k_{j=1}}$ 表示输出的模糊粒。标签模糊粒向量 $D = \{D_1, D_2, D_3, \dots, d_i\}$ ，其中 $d_i = \{s_j\}_{k_{j=1}}$ 表示标签模糊粒。粒损失函数定义如下。

(a). 绝对值损失函数:

$$L(G, D) = \frac{1}{l} \|G - D\|_1 = \frac{1}{l} \sum_{i=1}^l |g_i - d_i|. \quad (56)$$

两个粒度向量的绝对值损失函数度量是两个粒度向量之差的1-范数粒度均值。结果是一个粒。输出的粒向量和标记的粒向量之间的差异可以被测量。这个差异粒可以反向传播，以修正粒分类器中的加权粒。

(b). 平方损失函数:

$$L(G, D) = \frac{1}{l} \|G - D\|_2^2 = \frac{1}{l} \sum_{i=1}^l (g_i - d_i)^2. \quad (57)$$

两个粒向量的平方损失函数度量是两个粒向量之差的2-范数粒的平方均值，从而产生一个粒。

(c). 交叉熵损失函数:

$$L(G, D) = -\log(P) \cdot D = \sum_{i=1}^l \log(p_i) * d_i, \quad (58)$$

其中 $P = (P_1, P_2, \dots, p_i)$ 是softmax之后的概率粒向量。 PI 为概率粒，记为: $PI = \sum_{i=1}^l \log p_i$

3.5. 具有残差结构的模糊粒度深度卷积网络

粒级残差网络是参照残差网络的结构设计的。通过捷径机制将残差块添加到神经网络中。我们的网络结构类似于残差网络。但是，粒残差网络的输入是模糊粒，输出也是模糊粒。粒残差网络的结构图如图4所示。

在完成神经网络结构的构建后，我们接下来开始训练神经网络。神经网络的训练算法如算法1所示。

在完成网络训练后，我们将残差粒度神经网络应用于分类任务。粒度分类算法如下图所示。

4. 实验分析

为了验证模糊颗粒残差网络的分类性能，我们选择Cifar100 [42]和Tiny ImageNet [43]数据集进行对比实验。Cifar100由6万张大小为32*32的三通道彩色图像组成，其中5万张作为训练集，1万张作为测试集。Cifar100数据集的分类数为100类。微型Imagenet是

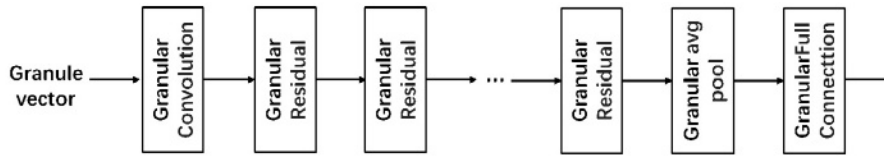


图4所示。粒度残差网络结构图。

Algorithm 1 Granular residual network learning algorithm

Input: The training set is $U = (X, Y)$, where x_i is an m -dimensional feature vector, $x_i \in X \subseteq R^m$, y_i is an l -dimensional label vector, which means it contains l categories, $y_i \in Y \subseteq R^l$, $i = 1, 2, \dots, n$.

Output: Granular weight matrix W and granular threshold b .

- (1) Randomly selected some samples as the reference sample set $S = P_1, P_2, \dots, P_k$.
- (2) The training set p is fuzzy granulated in the reference set P as $GT = \{(G(x_1), G(y_1)), (G(x_2), G(y_2)), \dots, (G(x_n), G(y_n))\}$, where the feature fuzzy granular vector is $G(x_i) = \{g_1(x_i), g_2(x_i), \dots, g_m(x_i)\}$, the label fuzzy granular vector is $G(y_i) = \{g_1(y_i), g_2(y_i), \dots, g_l(y_i)\}$.
- (3) Construct a multilayer granular neural network and randomly initialize the granular weights in the network.
- (4) For $x \in X$, input the characteristic fuzzy granular vector $G(x) = \{g_1(x), g_2(x), \dots, g_m(x)\}$ into the granular neural network for forward calculation; after the hidden layer calculation and softmax, the output granular vector is $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_l\}$.
- (5) Calculate the loss function $L(G(y), \alpha)$ of the output fuzzy granular vector and the label fuzzy granular vector, called a lossy fuzzy granule.
- (6) The lossy fuzzy granule is propagated back to the granular weight vector in the modified granular neural network according to the gradient direction.
- (7) Go to (4) until the loss function converges or the maximum number of iterations is reached.
- (8) Output granular neural network weight matrix W and granular threshold b , and reference sample set $S = \{P_1, P_2, \dots, P_k\}$.

Algorithm 2 Granular residual network classification algorithm

Input: The reference set $S = P_1, P_2, \dots, P_k$, granular neural network weight matrix W and granular threshold b , test samples x .

Output: The class of sample x .

- (1) The test sample x is granularized in the reference set S as $G(x) = \{g_1(x), g_2(x), \dots, g_m(x)\}$.
- (2) Input the granular vector $G(x)$ into the granular neural network for forward computation; after the hidden layer computation and softmax, output l fuzzy granules α_i .
- (3) Calculate the size of the fuzzy granule α_i by the formula $\alpha_i = \sum_{j=1}^k r_j$, where r_j is the granule kernel of α_i .
- (4) Calculate the probability of each class $\sigma_i = \frac{e^{\alpha_i}}{\sum_{t=1}^l e^{\alpha_t}}$.
- (5) Determine the test sample as the class with the highest probability.

从Imagenet数据集中选择。它包含200个类别，每个类别包含500张训练图像和50张测试图像，总共100000张训练图像和10000张测试图像。Tiny Imagenet中的图像分辨率为64*64，图像均为三通道彩色图像。数据集描述如表2所示。

表2数据集描述。

Dataset	Instance	Size	Class
Cifar100	60 000	32*32	100
Tiny Imagenet	110 000	64*64	200

本文的目的不是在数据集上取得最好的结果。相反，它是比较具有相似网络结构、相同网络层数、相同网络超参数设置的粒度神经网络和传统神经网络之间的差异。在实验部分，我们没有对数据集做任何数据增强，学习速率是默认设置的，在训练过程中没有手动调整。实验代码见此链接(<https://github.com/huangguagua555/Fuzzy-Granular-Residual-Network>)。

4.1. Cifar100

我们首先对样本集数据进行归一化。然后从样本集中随机选取 k 个样本作为参考集，对样本集进行粒度化处理。由于Cifar100数据集中图像的分辨率为32 * 32，因此所有卷积层的平面大小设置为3 * 3以有效地训练网络。

我们选择Resnet18和Resnet34与我们提出的模糊颗粒神经网络进行比较。模糊颗粒神经网络分别记为GResnet18和GResnet34。Resnet18与GResnet18结构相似，层数相同，Resnet34与GResnet34结构相似，层数相同。为了确保尽可能公平的比较，所有网络的优化器都是Aadm，学习率为0.001，beta1 = 0.9, beta2 = 0.999, epsilon = 1e-8。所有网络的Mini-batch被设置为128。

为了展示粒度神经网络的训练过程，我们展示了损失曲线和精度曲线。在训练过程中，Resnet34的训练集损失和测试集损失如图5所示。图6显示了Resnet34的训练集准确率和训练集准确率。从图5和图6可以看出，粒度神经网络可以很好地收敛，不会出现过拟合。

由于对数据进行模糊粒化选择的参考样本数量不同，神经网络的表现也有所不同。我们选取3 ~ 15个GResnet18网络的参考样本对训练数据进行模糊粒化。对于GResnet18网络，不同参考颗粒数对应的测试集正确率如图7所示。当参考颗粒数小于11时，分类正确率逐渐提高到60%左右。当参考颗粒数达到13时，随着参考颗粒数的增加，分类率在60.4%左右振荡。而当参考颗粒数达到15粒时，分类正确率下降到60.34%(见图7)。

我们还为GResnet34网络选择了3 ~ 15个参考样本进行模糊粒化，并对不同参考样本数量对应的测试集的正确分类率进行了比较

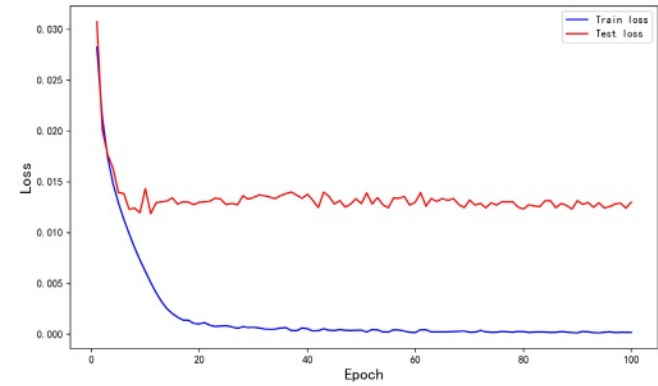


图5所示。GResnet34在Cifar100上的训练损耗和测试损耗。

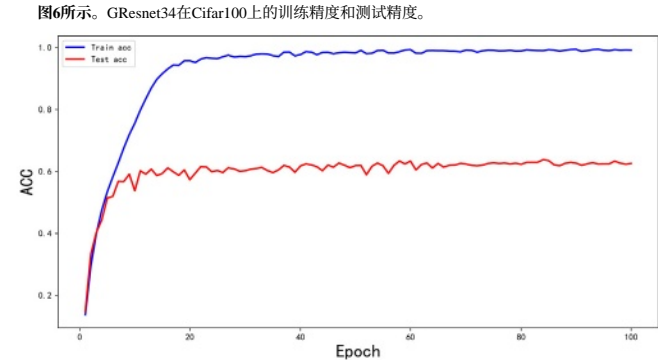


图6所示。GResnet34在Cifar100上的训练精度和测试精度。

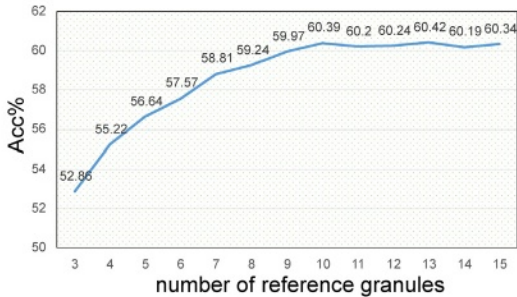


图7所示。参考颗粒数k对GResnet18准确度的影响。

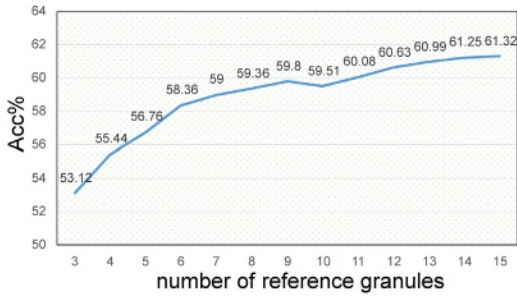


图8所示。参考颗粒数k对GResnet34准确度的影响。

颗粒如图8所示。随着颗粒化参考样本数量的增加，测试集正确率总体上也在增加。当参考样本粒数达到12~14时，正确率稳定。综上所述，测试集正确率与参考样本粒化数量基本呈正相关，测试集正确率将逐渐收敛

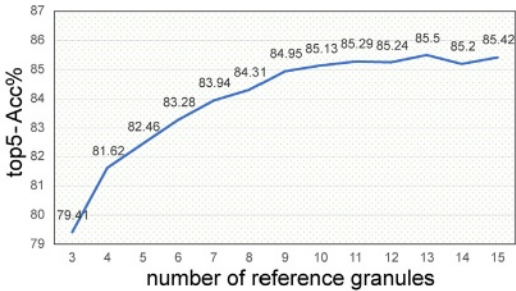


图9所示。参考颗粒数k对GResnet18中top5精度的影响

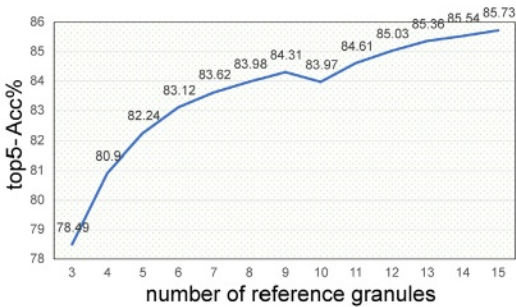


图10所示。参考颗粒数k对GResnet34中top5精度的影响。

表3

随机颗粒对Cifar100的影响。

Networks	Difference value	t = 1	t = 2	t = 3	t = 4	t = 5
GResnet18	Accuracy difference	0.31%	1.03%	0.95%	1.79%	1.49%
GResnet18	Top5 accuracy difference	0.12%	0.93%	0.87%	1.37%	1.5%
GResnet34	Accuracy difference	0.93%	1.37%	1.79%	1.38%	1.15%
GResnet34	Top5 accuracy difference	0.14%	0.1%	0.7%	1.11%	0.15%

当颗粒化的参考样本数量增加到足够大的尺寸时。

我们还给出了参考样本粒数对top-5准确率的影响。结果如图9和图10所示。top-5准确率也随着k的增加而增加。

通过一些实验来说明随机参考粒的作用。我们将k值设置为10。首先，我们冻结所有随机颗粒。之后，我们一个一个地改变随机颗粒。t表示被替换的随机颗粒的数量。表3显示了替换参考粒后的神经网络与原始神经网络精度的差异。对于GResnet18，当t=4时，原始网络与替代参考颗粒的网络准确度差最大，为1.79%。对于GResnet18，前5名的最大准确率差为1.37%。根据表3的实验结果，随机粒化对粒度神经网络的影响有限。

表4描述了Cifar100上GResnet34、GResnet18、Resnet34和Resnet18网络的一些训练设置和测试集准确性。所有模型都进行了5次重复实验，然后得到平均精度数据。所有网络的损失函数为交叉熵，所有网络的优化器为Adam，学习率为0.001。从表中可以看出，当网络隐藏层相同时，GResnet18正确率高于Resnet18，GResnet34正确率高于Resnet34。在测试集正确率上，只有18个隐藏层的GResnet18网络也高于有34个隐藏层的Resnet34网络。以及top5的准确率

表4

Cifar100中GResnet和Resnet的分类性能(平均值±std%)和设置。

Network	GResnet34	GResnet18	Resnet34	Resnet18
Number of hidden layers	34	18	34	18
Loss function	Cross entropy	Cross entropy	Cross entropy	Cross entropy
Optimizer	Adam	Adam	Adam	Adam
Learning Rate	0.001	0.001	0.001	0.001
Accuracy	61.36 ± 0.63%	60.28 ± 0.77%	54.54 ± 0.85%	54.34 ± 0.76%
Top5-accuracy	85.57 ± 0.75%	84.92 ± 0.64%	80.64 ± 0.72%	80.46 ± 0.72%
Top10-accuracy	91.51 ± 0.79%	91.01 ± 0.81%	88.36 ± 0.4%	88.33 ± 0.53%

表5

Cifar100的分类性能(平均值±std%)和颗粒残差网络变体和残差网络变体的设置。

Network	GResnet34	IResnet34	ReXnet	Resnext	Densenet	SeResnet
Loss function	Cross entropy	Cross entropy	Cross entropy	Cross entropy	Cross entropy	Cross entropy
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam
Learning Rate	0.001	0.001	0.001	0.001	0.001	0.001
Accuracy	64.51 ± 0.69%	59.8 ± 0.64%	63.11 ± 0.52%	58.01 ± 1.15%	61.95 ± 1.28%	57.72 ± 1.05%
Top5-accuracy	88.23 ± 0.74%	84.65 ± 0.65%	87.65 ± 0.72%	83.35 ± 0.69%	86.41 ± 0.82%	83.81 ± 0.7%
Top10-accuracy	93.31 ± 0.75%	91.24 ± 0.48%	92.39 ± 0.75%	90.34 ± 0.59%	92.38 ± 0.66%	90.9 ± 0.47%

表6

随机颗粒的影响在微小~

Network	Difference value	t = 1	t = 2	t = 3	t = 4	t = 5
GResnet34	Accuracy difference	0.13%	0.61%	0.45%	0.87%	0.9%
GResnet34	Top5 accuracy difference	0.62%	1.34%	1.09%	1.51%	1.5%
GResnet50	Accuracy difference	0.04%	0.26%	0.66%	0.09%	0.28%
GResnet50	Top5 accuracy difference	0.06%	0.16%	0.49%	0.45%	0.19%

GResnet18的top10精度也大于Resnet34和Resnet18。与Resnet18相比, Resnet34的性能几乎没有改善。与GResnet18相比, GResnet34有明显的性能改进。粒度残差网络具有更好的抑制网络退化的能力。

为了进一步展示粒度神经网络的有效性, 我们将粒度神经网络应用于残差神经网络的变体。参考IResnet的网络结构[44], 我们提出了粒状IResnet。我们将颗粒IResnet与剩余网络变体颗粒IResnet [44], ReXnet[45], 以及 Resnext [40], Densenet [46], SEnet[47]进行实验比较。相较于表5中具有相同超参数的残差网络变体, 粒度残差网络变体具有更好的分类性能。

4.2. 微小imagenet

首先我们对Tiny imagenet数据集进行了归一化。我们还选择了k个样本作为参考集, 并对样本集进行了粒度化。我们选择Resnet34和Resnet50与我们提出的模糊颗粒神经网络(分别表示为GResnet34和GResnet50)进行比较。卷积核大小和所有网络优化器设置与Cifar100数据集实验中相同。

图15和图12显示了GRes-net50的损耗和精度曲线。GResnet50在训练集和测试集上都能很好地学习。过度学习和欠学习不会出现。

对于GResnet34网络, 不同参考颗粒数量对应的测试集正确率如图13所示。随着参考颗粒数k的增加, 正确率逐渐增加。当k增加到13左右时正确率稳定下来, 正确率在48.5%左右。如图14所示, k对GResnet50正确率的影响与k对GResnet34的影响大致相同。随着k的增大, 准确率逐渐收敛到50.7%。

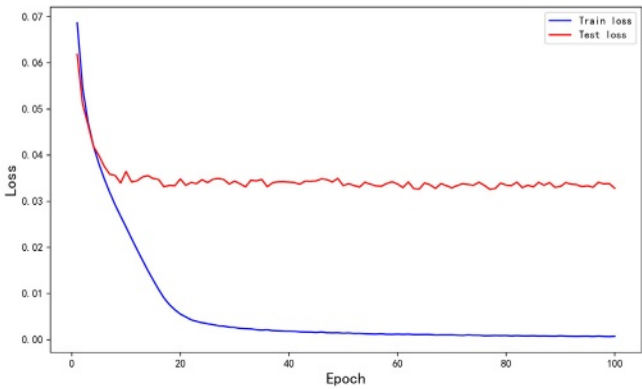
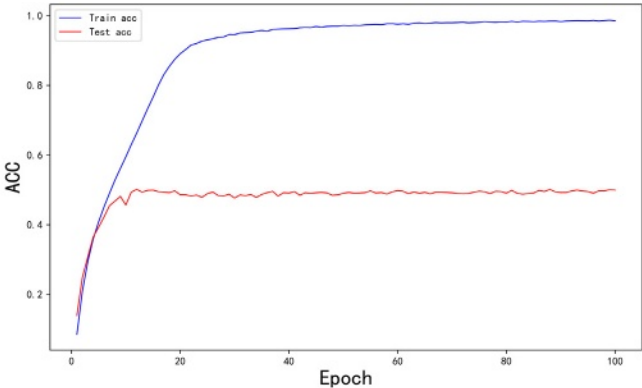


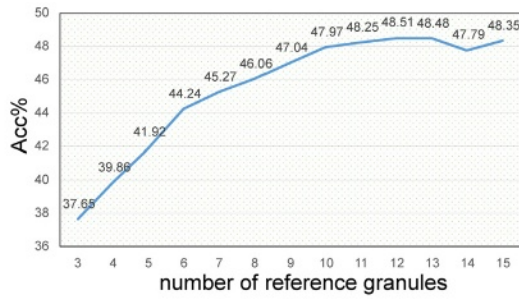
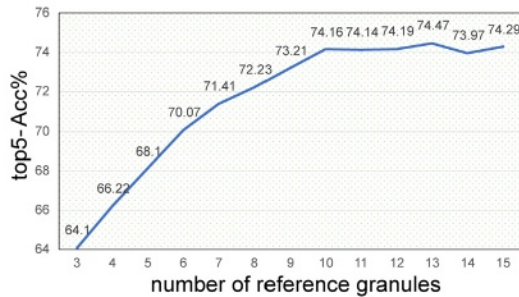
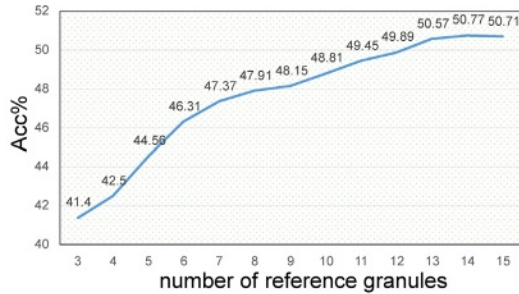
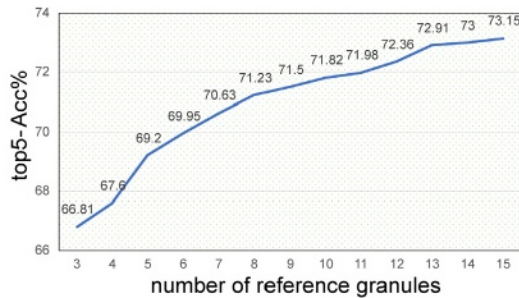
图11所示。GResnet50在Tiny Imagenet上的训练损耗和测试损耗。

图12所示。GResnet50在Tiny Imagenet上的训练精度和测试精度。



参考颗粒数对前5精度的影响如图15和图16所示。在GResnet34中, top-5的准确率随着k的增加而增加, 当k增加到10时, top-5的准确率在GResnet34中趋于平稳。然而, 在GResnet50中, 前5名的准确率随着k的增加而不断增长。

实验在一定程度上证明了粒度神经网络的性能随着参考集的增加而提高, 并最终达到收敛。在人类认知方面, 我们通过对比中不断向他人学习来认识自己的位置。它显示了粒度神经

图13所示。参考颗粒数 k 对GResnet34准确度的影响。图14所示。参考颗粒数 k 对GResnet50准确度的影响。图15所示。参考颗粒数 k 对GResnet34中top5精度的影响。图16所示。参考颗粒数 k 对GResnet50中top5精度的影响。

网络也可以通过增量学习不断强化自身。

同样，表6显示了当 $k = 10$ 时，GResnet34和GResnet50上不同随机颗粒对微小数据集的影响。当 $t = 2$ 时，GResnet50的精度公差和前5精度公差都最大。对于GResnet50，Top5精度最大差异小于1%。而且，最大精度差值也只有0.66%。

表7显示了在 tiny Imagenet 上 GResnet50、GResnet34、Resnet50和Resnet34网络的一些训练设

置和测试集准确性。所有模型运行5次重复。平均年龄精度在表7、8中得到。粒度残差神经网络的分类性能在相同层数下明显高于结构相似的残差网络。即使使用较少的颗粒残差神经网络层数，其在Tiny Imagenet数据集集中的分类精度也更高。虽然GResnet50的精度大于GResnet34，但GResnet34的top5精度和top10精度均大于GResnet50。

同样，基于IResnet50的残差网络变体对颗粒神经网络进行了修改。我们将改进的颗粒神经网络GIResnet50与Tiny Imagenet中的残差网络IResnet50[44]、ReXnet[45]、Resnext[36]、Densenet[46]、SEnet[47]的一些变体进行了比较。网络参数设置和结果如表8所示。所提出的GIResnet50在相同的超参数下，无论是在准确率、top5准确率、top10准确率方面都具有更好的分类性能。

总体而言，在相同的超参数下，所提出的粒度神经网络在残差网络和残差网络变体上都具有更好的性能。而且，不同的参考粒可以提供不同的视角，而传统的神经网络只学习当前样本的信息。因此，随着参考颗粒数 k 的增加，所提出的粒神经网络的性能不断提高。

5. 结论与讨论

本文提出了具有残差结构的模糊粒度深度卷积网络。通过随机选取参考系统，避免了全局粒化时间复杂度和空间复杂度过大的问题。由于粒具有良好的结构，它使神经元自然地并行。该算法从多个样本特征相似性的角度构建粒神经网络，与只关注当前样本的传统神经网络有着本质的不同。神经网络是一种多粒度的计算模型，粒度神经网络是一种多粒度、多角度的计算模型。这种多角度的学习模式，使得粒神经网络具有更好的泛化性能。更进一步，隐藏层较少的粒度神经网络可以实现隐藏层较多的普通神经网络的功能。我们还引入了神经网络的残差结构，对粒度神经网络进行了深化，使得粒度神经网络具有更好的表示能力。最后我们通过实验验证了残差粒度神经网络的可行性和有效性。

未来，粒神经网络将应用于更多领域，而不仅仅是分类任务。对于粒度神经网络的改进，有很多方向。例如，用邻域粒化代替粒化，会给粒神经网络带来不同的知识。将粒度神经网络与注意力机制相结合，也是一项有趣的工作。神经网络剪枝技术也可能降低粒度神经网络的复杂性。而且我们也将进一步构建更复杂、更深入的粒度网络。颗粒计算和深度学习可以更好地结合起来解决各种各样的复杂问题。在人类认知方面，我们通过在比较中不断向他人学习来认识自己的位置。

表7

Tiny Imagenet中GResnet和Resnet的分类性能(mean%±std%)和设置。

Network	GResnet50	GResnet34	Resnet50	Resnet34
Number of hidden layers	50	34	50	34
Loss function	Cross entropy	Cross entropy	Cross entropy	Cross entropy
Optimizer	Adam	Adam	Adam	Adam
Learning Rate	0.001	0.001	0.001	0.001
Accuracy	50.69 ± 0.81%	48.57 ± 0.79%	48.13 ± 0.95%	43.47 ± 0.49%
Top5-accuracy	72.78 ± 0.66%	75.14 ± 0.56%	73.07 ± 0.46%	69.2 ± 0.48%
Top10-accuracy	81.1 ± 0.87%	82.96 ± 0.5%	81.35 ± 0.28%	78.23 ± 0.45%

表8

Tiny Imagenet中颗粒残差网络变体和残差网络变体的分类性能(平均值%±std%)和设置。

Network	GIResnet50	IResnet50	ReXnet	Resnext	Densenet	SeResnet
Loss function	Cross entropy	Cross entropy	Cross entropy	Cross entropy	Cross entropy	Cross entropy
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam
Learning Rate	0.001	0.001	0.001	0.001	0.001	0.001
Accuracy	54.95 ± 0.53%	53.99 ± 0.56%	48.12 ± 0.52%	50.23 ± 1.15%	50.13 ± 0.83%	49.14 ± 0.42%
Top5-accuracy	78.81 ± 0.79%	77.43 ± 0.23%	73.85 ± 0.74%	74.12 ± 0.87%	74.15 ± 0.62%	73.53 ± 0.18%
Top10-accuracy	86.51 ± 0.55%	84.93 ± 0.37%	81.39 ± 0.65%	82.25 ± 0.91%	82.29 ± 0.65%	81.79 ± 0.46%

信用作者贡献声明

何林杰:概念化、方法论、验证、形式分析、写作-原稿、软件、项目管理、写作-评审和编辑。陈玉敏:方法论、软件、可视化、项目管理、写作-评审&编辑、资金获取。吴克寿:调查、监督、项目管理、写作-评审&编辑。

竞争利益声明

作者声明，他们没有已知的竞争性经济利益或个人关系，可能会影响这篇论文所报道的工作。

致谢。

本工作得到国家自然科学基金(No. 61976183)和中央地方科技发展专项(No. 2022L3029)的资助。

参考文献。

[1] L.A. Zadeh, Fuzzy Sets, *Inf. Control* 8 (3) (1965) 338–353.

[2] S.B. Roh, W. Pedrycz, T.C. Ahn, A design of granular fuzzy classifier, *Expert Syst. Appl.* 41 (15) (2014) 6786–6795.

[3] W. Pedrycz, H. Izakian, Cluster-centric fuzzy modeling, *IEEE Trans. Fuzzy Syst.* 22 (6) (2014) 1585–1597.

[4] J.R. Hobbs, Granularity, in: *Proc. of the IJCAI* (1985) 432–435.

[5] L.A. Zadeh, Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, *Fuzzy Sets and Systems* 90 (2) (1997) 111–127.

[6] W. Pedrycz, W. Homenda, Building the fundamentals of granular computing: A principle of justifiable granularity, *Appl. Soft Comput.* 13 (10) (2013) 4209–4218.

[7] T.Y. Lin, Data mining: Granular computing approach, *Lecture Notes in Comput. Sci.* 1574 (1999) 24–33.

[8] T.Y. Lin, L.A. Zadeh, Special issue on granular computing and data mining, *Int. J. Intell. Syst.* 19 (7) (2004) 565–566.

[9] Y. Y. Yao, Relational interpretations of neighborhood operators and rough set approximation operators, *Inf. Sci.* 111 (1998) 239–259.

[10] Y.Y. Yao, Information granulation and rough set approximation, *Int. J. Intell. Syst.* 16 (1) (2001) 87–104.

[11] M. Duoqian, F. Shidong, The calculation of knowledge granulation and its application, *Syst. Eng.-Theory Pract.* 22 (1) (2002) 48–56.

[12] D. Miao, N. Zhang, X. Yue, Knowledge reduction in interval-valued information systems, in: *8th IEEE International Conference on Cognitive Informatics*, 2009, pp. 320–327.

[13] R.R. Yager, Some learning paradigms for granular computing, in: *IEEE International Conference on Granular Computing*, 2006.

[14] Y.H. Qian, J.Y. Liang, C.Y. Dang, Incomplete multigranulation rough set, *IEEE Trans. Syst. Man Cybern. A* 40 (2010) 420–431.

[15] J. Qian, D.Q. Miao, Z.H. Zhang, et al., Parallel attribute reduction algorithms using MapReduce, *Inf. Sci.* 279 (2014) 671–690.

[16] M.A. Sanchez, O. Castillo, J.R. Castro, Information granule formation via the concept of uncertainty-based information with interval type-2 fuzzy sets representation and Takagi–Sugeno–Kang consequents optimized with Cuckoo search, *Appl. Soft Comput.* 27 (2015) 602–609.

[17] P. Melin, D. Sánchez, Multi-objective optimization for modular granular neural networks applied to pattern recognition, *Inform. Sci.* 460–461 (2018) 594–610.

[18] Y.M. Chen, N. Qin, W. Li, et al., Granule structures, distances and measures in neighborhood systems, *Knowl.-Based Syst.* 165 (2019) 268–281.

[19] D. Sánchez, P. Melin, O. Castillo, Comparison of particle swarm optimization variants with fuzzy dynamic parameter adaptation for modular granular neural networks for human recognition, *J. Intell. Fuzzy Syst.* 38 (3) (2020) 3229–3252.

[20] W. Li, Y.M. Chen, Y.P. Song, Boosted K-nearest neighbor classifiers based on fuzzy granules, *Knowl.-Based Syst.* 195 (2020) 105–606.

[21] G.P. Lin, J.Y. Liang, J.J. Li, A fuzzy multigranulation decision-theoretic approach to multi-source fuzzy information systems, *Knowl.-Based Syst.* 91 (2016) 102–113.

[22] T. He, W. Xu, Z. Lu, et al., Adaptive fuzzy logic energy management strategy based on reasonable SOC reference curve for online control of plug-in hybrid electric city bus, *IEEE Trans. Intell. Transp. Syst.* 19 (5) (2018) 1607–1617.

[23] Y.M. Chen, Q.X. Zhu, H.R. Xu, Finding rough set reducts with fish swarm algorithm, *Knowl.-Based Syst.* 81 (2015) 22–29.

[24] L.B. Cosme, W.M. Caminhas, M.F. Dangelo, R.M. Palhares, A novel fault-prognostic approach based on interacting multiple model filters and fuzzy systems, *IEEE Trans. Ind. Electron* 66 (1) (2018) 519–528.

[25] T. Lei, X. Jia, Y. Zhang, S. Liu, H. Meng, A.K. Nandi, Superpixel-based fast fuzzy C-means clustering for color image segmentation, *IEEE Trans. Fuzzy Syst.* 27 (9) (2019) 1753–1766.

[26] O. Castillo, et al., A generalized type-2 fuzzy granular approach with applications to aerospace, *Inform. Sci.* 354 (2016) 165–177.

[27] J.M. Mendel, P.P. Bonissone, Critical thinking about explainable AI (XAI) for rule-based fuzzy systems, *IEEE Trans. Fuzzy Syst.* 29 (12) (2021) 3579–3593.

[28] F. Rosenblatt, The perceptron: a probabilistic model for information storage and organization in the brain, *Psychol. Rev.* 65 (6) (1958) 386.

[29] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back propagating errors, *Nature* 323 (1986) 533–536.

[30] S. Hochreiter, J. Schmidhuber, Long short-term memory, *Neural Comput.* 9 (8) (1997) 1735–1780.

[31] Y. Lecun, L. Bottou, Gradient-based learning applied to document recognition, *Proc. IEEE* 86 (11) (1998) 2278–2324.

[32] G.E. Hinton, S. Osindero, Y.W. Teh, A fast learning algorithm for deep belief nets, *Neural Comput.* 18 (7) (2006) 1527–1554.

[33] Krizhevsky, Alex, Ilya. Sutskever, Geoffrey E. Hinton, Imagenet classification with deep convolutional neural networks, *Adv. Neural Inf. Process. Syst.* 25 (2012) 1097–1105.

[34] He. Kaiming, et al., Deep residual learning for image recognition, in: *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[35] S. Wu, S. Zhong, Y. Liu, Deep residual learning for image steganalysis, *Multimedia Tools Appl.* 77 (9) (2018) 10437–10453.

- [36] Tan. Runjie, et al., Color image demosaicking via deep residual learning, in: IEEE Int. Conf. Multimedia and Expo (ICME), vol. 2(4), 2017.
- [37] Yixin Du, Li Xin, Recursive deep residual learning for single image de-hazing, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018.
- [38] C. Liu, X. Liu, D.W.K. Ng, J. Yuan, Deep residual learning for channel estimation in intelligent reflecting surface-assisted multi-user communications, *IEEE Trans. Wirel. Commun.* 21 (2) (2022) 898–912.
- [39] H. Qiu, Q. Zheng, G. Memmi, J. Lu, B. Thuraishingham, Deep residual learning-based enhanced JPEG compression in the internet of things, *IEEE Trans. Industr. Inform.* 17 (3) (2021) 2124–2133.
- [40] Xie. Saining, et al., Aggregated residual transformations for deep neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017.
- [41] K. He, X. Zhang, S. Ren, S. Jian, Identity mappings in deep residual networks, in: European Conference on Computer Vision, Springer, Cham, 2016.
- [42] Alex Krizhevsky, Geoffrey Hinton, et al., Learning multiple layers of features from tiny images, 2009.
- [43] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., Imagenet large scale visual recognition challenge, *Int. J. Comput. Vis.* 115 (3) (2015).
- [44] I.C. Duta, et al., Improved Residual Networks for Image and Video Recognition, in: Proceedings of International Conference on Pattern Recognition, 2021.
- [45] D. Han, S. Yun, B. Heo, Y.J. Yoo, Rethinking Channel Dimensions for Efficient Model Design, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2021.
- [46] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely Connected Convolutional Networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017.
- [47] J. Hu, L. Shen, S. Albanie, G. Sun, E. Wu, Squeeze-and-excitation networks, *IEEE Trans. Pattern Anal. Mach. Intell.* 42 (8) (2020) 2011–2023.