混合曲率空间中的几何自适应元学习方法

高 志[1)] 武玉伟[2),1)] 贾云得[2),1)]

[1)](北京理工大学计算机学院智能信息技术北京市重点实验室 北京 100081)

[2)](深圳北理莫斯科大学广东省智能感知与计算重点实验室 广东 深圳 518172)

摘 要 元学习通过学习先验知识,能帮助模型快速适应新任务.在适应新任务的过程中,空间几何结构与数据几何结构的匹配程度对模型泛化起着重要作用.现实世界数据具有多样的非欧几何结构,例如自然语言具有非欧层级结构,人脸图像具有非欧环状结构等.已有研究表明,真实数据的非欧结构同黎曼流形的几何结构相匹配,从理论上提供了利用黎曼流形来建模数据的可行性.本文提出了混合曲率空间(mixed-curvature space)中的几何自适应元学习方法,利用多个混合曲率空间来表示数据,并生成与数据非欧结构相匹配的黎曼几何.本文构建了多混合曲率神经网络,将混合曲率空间的几何结构表示为曲率空间的曲率、数量和维度,由此通过梯度下降过程实现对数据非欧结构的几何自适应.本文进一步引入几何初始化生成策略和几何更新策略,通过少数几步迭代,空间几何结构即可快速匹配数据非欧结构,加速了梯度下降过程.本文在小样本分类和小样本回归等任务上进行了实验验证.与欧氏空间的元学习方法相比,本文方法在小样本分类任务上取得了约 3%的准确率提升,在小样本回归任务上将均方误差减少了一半,验证了本文方法的有效性.

关键词 元学习;几何自适应;混合曲率空间;黎曼流形

中图法分类号 TP391 **DOI 号** 10.11897/SP.J.1016.2024.02289

Geometry-Adaptive Meta-Learning in Mixed-Curvature Spaces

GAO Zhi[1)] WU Yu-Wei[2),1)] JIA Yun-De[2),1)]

[1)](*Beijing Key Laboratory of Intelligent Information Technology, School of Computer Science & Technology, Beijing Institute of Technology, Beijing* 100081)

[2)](*Guangdong Laboratory of Machine Perception and Intelligent Computing, Shenzhen MSU-BIT University, Shenzhen, Guangdong* 518172)

Abstract Meta-learning has shown effectiveness in helping learning models quickly adapt to new tasks by learning prior knowledge. In the process of adaptation to new tasks, the matching degree between the geometric structure of space and the geometric structure of data plays an important role in the generalization ability of the model. In many practical applications, data has diverse non-Euclidean structures. For example, natural language has non-Euclidean hierarchical structures, and face images have non-Euclidean cyclical structures. Existing research has shown that the geometric structure of Riemannian manifolds matches the non-Euclidean structures of real-world data, providing theoretical feasibility for modeling data using Riemannian manifolds. In this paper, we propose a geometry-adaptive meta-learning method in mixed-curvature spaces, which uses multiple mixed-curvature spaces to model data and produces matching Riemannian geometry for non-Euclidean structures. We build a multi-mixed-curvature neural network that represents the geometry of mixed-curvature space as curvature, number, and dimensionality of the

收稿日期:2023-05-21;在线发布日期:2024-07-04.本课题得到国家自然科学基金(62172041,62176021)、深圳市自然科学基金面上项目(JCYJ20230807142703006)、广东省教育厅普通高校重点科研平台和项目(2023ZDZX1034)资助.**高 志**,博士,主要研究方向为计算机视觉、机器学习和黎曼几何.E-mail:gaozhi_2017@126.com.**武玉伟**(通信作者),博士,长聘副教授,博士生导师,中国计算机学会(CCF)会员,主要研究领域为计算机视觉和机器学习.E-mail:wuyuwei@bit.edu.cn.**贾云得**,博士,教授,博士生导师,中国计算机学会(CCF)会员,主要研究领域为计算机视觉、计算认知和智能系统.

curvature spaces, through which the geometry adaptation to non-Euclidean structures is achieved via a gradient descent process. We further introduce a geometry initialization generation scheme and geometry updating scheme. Through only a few optimization steps, the geometric structure of the underlying space can quickly match non-Euclidean structures of data, accelerating the gradient descent process. We conduct experiments on few-shot classification, few-shot regression, and image completion to evaluate the effectiveness of our method. Compared with meta-learning methods in Euclidean space, our method improves the accuracy by 3% in few-shot classification tasks, and reduces mean square error by half in few-shot regression tasks, showing the effectiveness of our method.

Keywords meta-learning; geometry adaptation; mixed-curvature space; Riemannian manifold

1 引 言

元学习的核心思想是在已知任务上总结先验知识,教会模型如何高效学习新任务,以缓解深度学习泛化性不强、需要大量标注数据以及繁琐训练过程的问题[1-3].元学习方法通常包含两个模型:基础学习模型和元学习模型.通过训练元学习模型学习先验知识,帮助基础学习模型快速适应新任务.目前主流的元学习方法大致分为三类:基于优化的元学习方法、基于度量的元学习方法和基于模型的元学习方法[4].本文方法属于基于优化的元学习方法.该类方法展现了较好的灵活性,在图像分类、强化学习、语义分割、目标检测和 3D 视觉[5-7]等多个应用中表现出优异的性能.

目前大多数基于优化的元学习方法假设数据都遵循欧氏结构,并在这一前提下探究模型如何适应新任务[8-10].然而,现实世界中的数据结构远比欧氏结构复杂[11],例如自然语言和细粒度图像具有非欧层级结构[12],人脸图像和一些知识图谱数据具有非欧环状结构[13].此外,一些任务的数据具有非均匀的非欧结构,即某个区域具有环状结构,而另一区域具有层级结构[14].显然,欧氏空间的几何结构与数据的非欧几何结构不匹配.将这些复杂的数据强行嵌入欧氏空间,会扭曲数据的真实结构,削弱了数据表达的丰富性和准确性.这种失真不仅限制了模型有效捕捉数据间复杂关系的能力,还严重影响其泛化至新任务的性能,从而制约了模型的适应范围.

已有研究表明,真实数据所蕴含非欧结构同黎曼流形结构高度契合[15],为利用黎曼流形来建模数据提供了理论依据.使用与非欧结构相匹配的黎曼几何可以保持数据的几何结构,增强数据表示的表达力[16].例如,与非欧层级结构相匹配的是双曲几何,双曲空间的容量随着半径增长而呈指数增长,而欧氏空间的容量随半径增长呈线性增长,这使得双曲几何在建模层级结构中数量呈指数增长的叶子结点时更具优势[17].再如,与非欧环状结构相匹配的是球面几何,使用球面几何可以更准确地度量环状数据间的相似性[18].

混合曲率空间(mixed-curvature space)是常见的黎曼流形[13-14,18-19],由常曲率空间(constant curvature space)的笛卡儿积组成,通常建模为常曲率空间表示的串联.混合曲率空间的几何结构取决于常曲率空间的数量、维度和曲率[20-22].通过更新常曲率空间的数量、维度和曲率,混合曲率空间可以匹配多种多样的非欧结构[17].真实场景中的任务常常面临数据匮乏,数据结构复杂多样且不可提前预知的特点.混合曲率空间以几何结构灵活的建模方式为模型与数据的匹配提供了一条可行的途径.例如,使用带有正曲率的常曲率空间有助于建模人脸图像和一些知识图谱数据的非欧环状结构,使用带有负曲率的常曲率空间有助于建模自然语言和细粒度图像的非欧层级结构.因此,本文研究混合曲率空间中的几何自适应元学习方法,该方法学习混合曲率空间几何结构与数据几何结构间的先验知识,进而调整常曲率空间的数量、维度和曲率,生成与新任务数据相匹配的空间几何结构,提升泛化性能.

实现混合曲率空间中的几何自适应面临着两个挑战.第一个挑战是如何构建基础学习模型,以保持混合曲率空间的几何结构.现有元学习方法所使用的模型(例如各种卷积神经网络)会不可避免地破坏混合曲率空间的几何结构,降低模型的泛化能力.第二个挑战是如何以可微分的方式动态更新常曲率空间的数量和维度.鉴于元学习的训练过程通常计算

梯度，而常曲率空间的数量与维度作为离散变量，阻碍了直接采用传统意义上的微分策略进行优化，这一矛盾成为了技术突破的关键瓶颈.

为了解决第一个挑战，本文搭建了黎曼流形中的基础学习模型：多混合曲率神经网络. 该神经网络使用多个混合曲率空间来建模数据，这些混合曲率空间由不同数量、不同维度和不同曲率的常曲率空间组成. 通过为混合曲率空间分配不同的权重以匹配多样的非欧几何结构. 为了解决第二个挑战，本文引入了几何初始化生成策略和几何更新策略，通过可微分的梯度下降过程对混合曲率空间的权重和曲率进行更新. 几何初始化生成策略生成梯度下降过程中合适的权重和曲率初始化；几何更新策略输出梯度下降过程的学习率和搜索方向. 得益于这一系列设计，经过少数几步迭代，本文模型的混合曲率空间即可快速适应数据的非欧结构. 本文在小样本分类、小样本回归和图像填充任务上进行了实验验证. 与欧氏空间中的元学习方法相比，在小样本分类任务上本文方法实现了约 3% 的准确率增长，在小样本回归任务上将均方误差减少至原来的一半，验证了本文方法的有效性.

2　相关工作

2.1　元学习

已有元学习方法大致分为三类：基于度量的元学习方法、基于模型的元学习方法和基于优化的元学习方法. 基于度量的元学习方法通过学习鲁棒的嵌入空间和距离度量实现对新任务的快速适应. 基于度量的方法为所有任务学习公共嵌入空间[23-24]，但限制了模型的适应能力. 为了解决这个问题，近期不少工作研究任务自适应嵌入空间，包括学习任务自适应的特征变换[25]、部署任务自适应的目标函数[26]、生成任务自适应的类别原型[27]等. 最近，Khrulkov 等人[12]和 Qi 等人[28]提出了黎曼流形中基于度量的元学习方法，学习鲁棒的非欧距离度量. 然而，由于这两个方法对距离度量的依赖，它们仅能用于分类任务[23]. 与这两个方法不同，本文方法属于基于优化的元学习方法，可以灵活地应用于分类、回归和强化学习等多种任务.

基于模型的元学习方法将自适应过程建模为任务表示的学习过程和黑盒模型的前向传播过程，黑盒模型输出为自适应后的模型参数. Santoro 等人[29]利用梯度信息作为任务表示来预测神经网络的参数. Xu 等人[30]引入了编码器-解码器架构来生成无限维任务表示. Mishra 等人[31]使用时间卷积和注意力机制来提高基于模型的元学习方法的记忆能力. Zhen 等人[32]利用任务表示来生成特定的核函数，并在多个任务上显示了先进的性能. 与上述基于模型的元学习方法相比，因为本文方法无需对已知任务的表示和参数进行存储，因此本文方法的计算资源消耗更少.

基于优化的元学习方法主要位于欧氏空间，通过基础学习模型的优化过程实现对新任务的快速适应. 一些方法致力于好的基础学习模型的参数初始化，其中最著名的方法是模型无关元学习方法(Model-Agnostic Meta-Learning，MAML)[1]，在多个计算机视觉、机器学习领域得到了成功应用. 许多研究者提出了 MAML 改进的方法[2,5,10]，旨在降低 MAML 方法的计算复杂性并克服过拟合问题. 此外，一些工作表明，基础学习模型的更新过程也可以被建模为元学习问题. 这些工作使用神经网络作为优化器，并训练此神经网络实现高效的参数优化[33-34]. 与上述方法不同的是，本文方法关注实际应用中更为常见的非欧结构数据，通过混合曲率空间，实现了对非欧结构数据的几何自适应，具有更广的应用场景和更强的泛化能力. 最近，Gao 等人[21]提出了曲率自适应的元学习方法. 与之相比，本文方法不仅生成自适应的曲率，还生成自适应的曲率空间数量和维度，实现了更灵活的几何自适应，以更小的数据失真来匹配非欧数据.

2.2　混合曲率空间中的学习算法

混合曲率空间是由多个常曲率空间组成的积流形. 常曲率空间作为积流形中的子流形，具有建模非欧结构数据的能力. 曲率是常曲率空间中的重要概念，表示该空间与平坦欧氏空间的差异[19,35]. 正曲率定义了适合于环状结构的球面空间，负曲率定义了适合于层级结构的双曲空间. 一些方法聚焦于将欧氏空间中的算法与模型推广至常曲率空间，例如核方法[16]、多层感知器[35]、降维算法[36]、数据增强[37]、图神经网络[38-39]和生成式模型[40].

实际应用中的数据几何结构比较复杂，数据通常具有非均匀的非欧结构. 相比于单一的空间几何结构(例如单一的球面空间)，由多个常曲率空间组成的混合曲率空间可以更好地匹配复杂的非欧结构. 比如，一些方法通过实验试错的方式来确定混合曲率空间的曲率[18-19,21]，获得与数据非欧结构相匹配的混合曲率空间几何. 最近的一些方法将混合曲率空

间的几何结构看作模型的可学习参数,通过监督学习的方式获得混合曲率空间的几何结构[13-14,20].上述方法都需要大量的时间消耗或数据标注.不同于上述方法,本文方法通过学习先验知识,可以在数据量较少的情况下快速适应复杂的非欧结构数据.

2.3 黎曼流形中的神经网络

目前黎曼流形中的神经网络可以分为三类.第一类神经网络使用欧氏空间中的骨干网络,并在骨干网络顶部添加若干流形运算,以获得数据的流形表示[12,28].在这种情况下,欧氏骨干网络仍然可能破坏数据的非欧结构.第二类神经网络专门为流形特征[41]设计.换句话说,第二类神经网络不涉及处理原始数据的网络架构(例如处理图像数据的卷积运算),而仅具有处理流形特征的网络架构(例如线性投影运算).为了解决这两个问题,第三类神经网络应运而生,即黎曼流形中的端到端神经网络.第三类神经网络研究黎曼流形中的几何保持操作来处理原始数据并保证数据的流形结构不被破坏[38].多混合曲率神经网络属于黎曼流形中的第三类神经网络,可以端到端地处理输入数据.但与已有的神经网络相比,本文所搭建的多混合曲率神经网络使用多个黎曼流形来建模数据,而不是单一空间,能够更灵活地匹配数据中复杂的非欧结构.

3 数学背景

3.1 常曲率空间

常曲率空间是一种平滑的黎曼流形.维度为 d、曲率为 K 的常曲率空间通常表示为 $\mathbb{C}_K^d$.曲率 K 的符号定义了三种类型的空间.负曲率对应着双曲空间,我们选择使用双曲空间中的庞加莱圆盘模型[42](Poincar'e ball);零曲率对应着欧氏空间;正曲率对应着球面空间,我们选择使用球面空间中的投影球模型[43](projected sphere model).在本文中,常曲率空间定义如下:

$$\mathbb{C}_K^d=\begin{cases}\mathbb{D}_K^d=\left\{\boldsymbol{x}\in\mathbb{R}^{d+1}\mid\langle\boldsymbol{x},\boldsymbol{x}\rangle_2=\dfrac{1}{K}\right\}, & K>0\\ \mathbb{E}^d=\mathbb{R}^d, & K=0\\ \mathbb{H}_K^d=\left\{\boldsymbol{x}\in\mathbb{R}^{d+1}\mid\langle\boldsymbol{x},\boldsymbol{x}\rangle_2<\dfrac{1}{K}\right\}, & K<0\end{cases}.$$

常曲率空间中以点 $\boldsymbol{u}\in\mathbb{C}_K^d$ 为切点的切空间表示为 $T_{\boldsymbol{u}}\mathbb{C}_K^d$.切空间是欧氏空间,可以在切空间中直接执行欧氏空间的操作.本文使用陀螺矢量空间[42](gyrovector space)为常曲率空间提供基础操作.

(1)常曲率空间中的向量加法.对于常曲率空间中的两点 $\boldsymbol{x},\boldsymbol{y}\in\mathbb{C}_K^d$,其加法由莫比乌斯加法(Möbius addition)定义,

$$\boldsymbol{x}\oplus_K\boldsymbol{y}=\frac{(1-2K\langle\boldsymbol{x},\boldsymbol{y}\rangle_2-K\|\boldsymbol{y}\|^2)\boldsymbol{x}+(1+K\|\boldsymbol{x}\|^2)\boldsymbol{y}}{1-2K\langle\boldsymbol{x},\boldsymbol{y}\rangle_2+K^2\|\boldsymbol{x}\|^2\|\boldsymbol{y}\|^2}.$$

(2)常曲率空间中的指数对数运算.指数运算 $\exp_{\boldsymbol{u}}^K(\cdot):T_{\boldsymbol{u}}\mathbb{C}_K^d\to\mathbb{C}_K^d$ 和对数运算 $\log_{\boldsymbol{u}}^K(\cdot):\mathbb{C}_K^d\to T_{\boldsymbol{u}}\mathbb{C}_K^d$ 可以实现数据在常曲率空间与切空间之间的变换,

$$\begin{cases}\exp_{\boldsymbol{u}}^K(\boldsymbol{q})=\boldsymbol{u}\oplus_K\left(\tan_K\left(\sqrt{|K|}\dfrac{\lambda_{\boldsymbol{u}}^K\|\boldsymbol{q}\|}{2}\right)\dfrac{\boldsymbol{q}}{\sqrt{|K|}\cdot\|\boldsymbol{q}\|}\right)\\ \log_{\boldsymbol{u}}^K(\boldsymbol{x})=\\ \dfrac{2}{\sqrt{|K|}\lambda_{\boldsymbol{u}}^K}\tan_K^{-1}(\sqrt{|K|}\cdot\|-\boldsymbol{u}\oplus_K\boldsymbol{x}\|)\dfrac{-\boldsymbol{u}\oplus_K\boldsymbol{x}}{\|-\boldsymbol{u}\oplus_K\boldsymbol{x}\|}\end{cases}.$$

如果曲率 K 大于0,则 $\tan_K(\cdot)=\tan(\cdot)$,否则 $\tan_K(\cdot)=\tanh(\cdot)$;同理,如果 K 大于0,则 $\tan_K^{-1}(\cdot)=\tan^{-1}(\cdot)$,否则 $\tan_K^{-1}(\cdot)=\tanh^{-1}(\cdot)$.

(3)常曲率空间中的测地线距离.测地线距离表示黎曼流形中两个点之间的最短距离.对于常曲率空间中的两点 $\boldsymbol{x},\boldsymbol{y}\in\mathbb{C}_K^d$,其测地线距离定义为

$$\psi_K(\boldsymbol{x},\boldsymbol{y})=\frac{2}{\sqrt{|K|}}\tan_K^{-1}(\sqrt{|K|}\cdot\|-\boldsymbol{x}\oplus_K\boldsymbol{y}\|).$$

(4)常曲率空间中的正交投影.正交投影(orthogonal projection)$\mathrm{proj}_{\boldsymbol{u}}^K$ 将一个任意空间中的向量 $\boldsymbol{z}$ 投影至切空间 $T_{\boldsymbol{u}}\mathbb{C}_K^d$ 上,

$$\mathrm{proj}_{\boldsymbol{u}}^K(\boldsymbol{z})=\begin{cases}\boldsymbol{z}-\langle\boldsymbol{u},\boldsymbol{z}\rangle_2\boldsymbol{u}, & K\geq0\\ (1/(\lambda_{\boldsymbol{u}}^K)^2)\boldsymbol{z}, & K<0\end{cases}.$$

3.2 混合曲率空间

混合曲率空间是由多个常曲率空间组成的笛卡儿积[18-19],

$$\mathcal{M}:=\times_{j=1}^m\mathbb{C}_{K_j}^{d_j},$$

其中混合曲率空间 $\mathcal{M}$ 共由 m 个常曲率空间组成,$\mathbb{C}_{K_j}^{d_j}$ 表示组成混合曲率空间的第 j 个常曲率空间.混合曲率空间中的点 $\boldsymbol{x}\in\mathcal{M}$ 通常被表示为向量的拼接,$\boldsymbol{x}=(\boldsymbol{x}^1,\cdots,\boldsymbol{x}^m)$,其中 $\boldsymbol{x}^j\in\mathbb{C}_{K_j}^{d_j}$.所有常曲率空间的曲率表示为 $\mathcal{K}=\{K_1,\cdots,K_m\}$.同样地,混合曲率空间 $\mathcal{M}$ 的切空间 $T_{\boldsymbol{u}}\mathcal{M}$ 被定义为常曲率空间切空间的笛卡儿积,

$$T_{\boldsymbol{u}}\mathcal{M}:=\times_{j=1}^m T_{\boldsymbol{u}^j}\mathbb{C}_{K_j}^{d_j},$$

其中 $\boldsymbol{u}=(\boldsymbol{u}^1,\cdots,\boldsymbol{u}^m)\in\mathcal{M}$ 表示 $T_{\boldsymbol{u}}\mathcal{M}$ 的切点.为了方便,本文设置某一混合曲率空间中的所有常曲率空间具有相同的维度 d,则混合曲率空间的维度为 $D=dm$.混合曲率空间的几何结构 $\boldsymbol{g}$ 定义为常曲率空间的曲率 $\mathcal{K}$、维度 d 和数量 m.

通过设置常曲率空间的数量 m、维度 d 和曲率 $\mathcal{K}$，混合曲率空间可以匹配多种多样的非欧结构. 本文方法学习先验知识，赋予模型从给定任务的少量数据中估计 m、d 和 $\mathcal{K}$，以应对真实任务中数据匮乏、数据结构复杂多样且不可提前预知的特点，生成与新任务相匹配的空间几何结构，提升模型的泛化性能. 具体地，本文方法构建了多混合曲率神经网络，使用多个混合曲率空间来建模数据. 这些混合曲率空间由不同数量、不同维度和不同曲率的常曲率空间组成，通过为混合曲率空间分配不同的权重以调整曲率空间的数量 m 和维度 d；本文方法引入了几何初始化生成策略和几何更新策略，给定新任务，通过梯度下降过程对曲率进行调整. 具体细节请查阅方法章节.

(1) 混合曲率空间中的指数对数运算. 混合曲率空间中的指数运算 $\mathrm{Expm}_{\boldsymbol{u}}^{\mathcal{K}}(\cdot):T_{\boldsymbol{u}}\mathcal{M}\to\mathcal{M}$ 和对数运算 $\mathrm{Logm}_{\boldsymbol{u}}^{\mathcal{K}}(\cdot):\mathcal{M}\to T_{\boldsymbol{u}}\mathcal{M}$ 实现了数据在混合曲率空间与切空间之间的变换，

$$\begin{cases}\mathrm{Expm}_{\boldsymbol{u}}^{\mathcal{K}}(\boldsymbol{q})=(\exp_{\boldsymbol{u}^1}^{K_1}(\boldsymbol{q}^1),\cdots,\exp_{\boldsymbol{u}^m}^{K_m}(\boldsymbol{q}^m))\\ \mathrm{Logm}_{\boldsymbol{u}}^{\mathcal{K}}(\boldsymbol{x})=(\log_{\boldsymbol{u}^1}^{K_1}(\boldsymbol{x}^1),\cdots,\log_{\boldsymbol{u}^m}^{K_m}(\boldsymbol{x}^m))\end{cases}.$$

(2) 混合曲率空间中的平方距离. 混合曲率空间中的平方距离定义为常曲率空间中的平方距离之和，

$$\Psi_{\mathcal{K}}^2(\boldsymbol{x},\boldsymbol{y})=\sum_{j=1}^{m}\psi_{K_j}^2(\boldsymbol{x}^j,\boldsymbol{y}^j),$$

其中 $\psi_{\mathcal{K}}^2(\cdot)$ 表示常曲率空间 $\mathbb{C}_{K_j}^{d_j}$ 上的平方距离.

(3) 混合曲率空间中的正交投影. 混合曲率空间中的正交投影 $\mathrm{Proj}_{\boldsymbol{u}}^{\mathcal{K}}$ 将任意空间中的向量 $\boldsymbol{z}$ 投影到混合曲率空间的切空间 $T_{\boldsymbol{u}}\mathcal{M}$ 中，

$$\mathrm{Proj}_{\boldsymbol{u}}^{\mathcal{K}}(\boldsymbol{z})=(\mathrm{proj}_{\boldsymbol{u}^1}^{K_1}(\boldsymbol{z}^1),\cdots,\mathrm{proj}_{\boldsymbol{u}^m}^{K_m}(\boldsymbol{z}^m)).$$

4　方　法

4.1　问题定义

在元学习中，考虑任务分布 $p(\mathcal{T})$，任务 $\mathcal{T}_i$ 采样自 $p(\mathcal{T})$，即 $\mathcal{T}_i\sim p(\mathcal{T})$. 任务 $\mathcal{T}_i$ 由包含训练数据的支持集 $\mathcal{D}_i^s$ 和包含测试数据的查询集 $\mathcal{D}_i^q$ 组成. 元学习的目标是训练元学习模型，从任务分布 $p(\mathcal{T})$ 中学习先验知识，帮助基础学习模型快速适应新任务.

本文方法使用多个混合曲率空间来建模数据，构建了多混合曲率神经网络 $\mathcal{N}_{\boldsymbol{g},\boldsymbol{\theta}}$ 作为基础学习模型，其中 $\boldsymbol{g}$ 表示神经网络中的空间几何结构，$\boldsymbol{\theta}$ 表示神经网络的参数. 本文引入几何自适应优化算法 $\mathcal{A}_\psi$，包括几何初始化生成策略和几何更新策略，将带有公共初始化 $\boldsymbol{g}$ 和 $\boldsymbol{\theta}$ 的模型 $\mathcal{N}_{\boldsymbol{g},\boldsymbol{\theta}}$ 通过梯度下降过程优化为任务特定的模型 $\mathcal{N}_{\boldsymbol{g}_i',\boldsymbol{\theta}_i'}$

$$\mathcal{N}_{\boldsymbol{g}_i',\boldsymbol{\theta}_i'}=\mathcal{A}_\psi(\mathcal{N}_{\boldsymbol{g},\boldsymbol{\theta}},\mathcal{D}_i^s)\tag{1}$$

其中 $\boldsymbol{g}_i'$ 与 $\boldsymbol{\theta}_i'$ 分别表示任务特定的空间几何结构和模型参数. $\mathcal{A}_\psi$ 将公共初始化 $\boldsymbol{g}$ 转换为任务特定的几何初始化，并生成梯度下降过程中的学习率和更新方向，使得多混合曲率神经网络 $\mathcal{N}_{\boldsymbol{g},\boldsymbol{\theta}}$ 可以快速适应于各种非欧结构.

因此，本文方法的元学习模型包括多混合曲率神经网络的公共几何初始化 $\boldsymbol{g}$、几何自适应优化算法的参数 ψ 以及参数初始化 $\boldsymbol{\theta}$，可以建模为

$$\arg\min_{\boldsymbol{g},\boldsymbol{\theta},\psi}\mathbb{E}_{\mathcal{T}_i\sim p(\mathcal{T})}[\mathcal{L}(\mathcal{A}_\psi(\mathcal{N}_{\boldsymbol{g},\boldsymbol{\theta}},\mathcal{D}_i^s),\mathcal{D}_i^q)]\tag{2}$$

式(2)利用双层循环机制训练元学习模型. 在内层循环中，$\mathcal{A}_\psi$ 利用支持集数据 $\mathcal{D}_i^s$ 将多混合曲率神经网络 $\mathcal{N}_{\boldsymbol{g},\boldsymbol{\theta}}$ 快速适应于任务 $\mathcal{T}_i$. 在外层循环中，利用查询集数据 $\mathcal{D}_i^q$ 计算任务特定模型 $\mathcal{N}_{\boldsymbol{g}_i',\boldsymbol{\theta}_i'}$ 的损失 $\mathcal{L}$，并更新 $\boldsymbol{g}$、ψ 和 $\boldsymbol{\theta}$.

4.2　多混合曲率神经网络

多混合曲率神经网络使用多个混合曲率空间来建模数据，并为不同的混合曲率空间分配不同的权重. 假设多混合曲率神经网络的第 i 层使用 r 个混合曲率空间：$\mathcal{M}_1,\mathcal{M}_2,\cdots,\mathcal{M}_r$. 此 r 个混合曲率空间具有相同的维度，$D_1=D_2=\cdots=D_r$. 同时，为了简单易操作，设置同一个混合曲率空间中的常曲率空间拥有相同的维度，即第 n 个混合曲率空间 $\mathcal{M}_n$ 由 m_n 个维度为 d_n 的常曲率空间组成，

$$\mathcal{M}_n:=\mathbb{C}_{K_n^1}^{d_n}\times\mathbb{C}_{K_n^2}^{d_n}\times\cdots\times\mathbb{C}_{K_n^{m_n}}^{d_n}\tag{3}$$

混合曲率空间的维度为 $D_n=m_nd_n$. 值得注意的是，不同混合曲率空间中常曲率空间的数量和维度是不同的，即对于 $n\neq n'$，有 $m_n\neq m_{n'}$，$d_n\neq d_{n'}$.

混合曲率空间具有非欧几何结构，因此无法使用传统的神经网络架构，例如欧氏空间中的全连接层和卷积块. 考虑到流形的切空间是欧氏空间，可以利用指数映射和对数映射将传统的神经网络推广至混合曲率空间. 在神经网络的第 i 层，输入是数据在 r 个混合曲率空间 $\{\mathcal{M}_n\}_{n=1}^r$ 中的特征表示. 首先使用 r 个对数映射 $\{\mathrm{Logm}_{\boldsymbol{u}_n}^{\mathcal{K}_n}\}_{n=1}^r$ 将输入特征从 r 个混合曲率空间 $\{\mathcal{M}_n\}_{n=1}^r$ 映射至其对应的切空间 $\{T_{\boldsymbol{u}_n}\mathcal{M}_n\}_{n=1}^r$，$\boldsymbol{u}_n$ 表示第 n 个混合曲率空间中的切点. 接着，对不同切空间中的特征使用相同的变换操作(例如线性映射和非线性激活)进行处理. 考虑到处理后的特征位于任意的空间，可以直接将不同混合曲率空间的结果加权组合，加权后的结果被映射至神经网

络第 $i+1$ 层的 r 个混合曲率空间 $\{\mathcal{M}'_n\}_{n=1}^{r}$. 为此,使用 r 个正交投影操作 $\{\mathrm{proj}_{\boldsymbol{u}'_n}^{\mathcal{K}'_n}\}_{n=1}^{r}$ 将特征映射到切空间 $\{T_{\boldsymbol{u}'_n}\mathcal{M}'_n\}_{n=1}^{r}$ 中,其中 $\boldsymbol{u}'_n\in\mathcal{M}'_n$ 是神经网络第 $i+1$ 层中第 n 个混合曲率空间的切点,并使用 r 个指数映射 $\{\mathrm{Expm}_{\boldsymbol{u}'_n}^{\mathcal{K}'_n}\}_{n=1}^{r}$ 将切空间 $\{T_{\boldsymbol{u}'_n}\mathcal{M}'_n\}_{n=1}^{r}$ 中的结果映射至 $\{\mathcal{M}'_n\}_{n=1}^{r}$. 上述过程如图 1 所示.

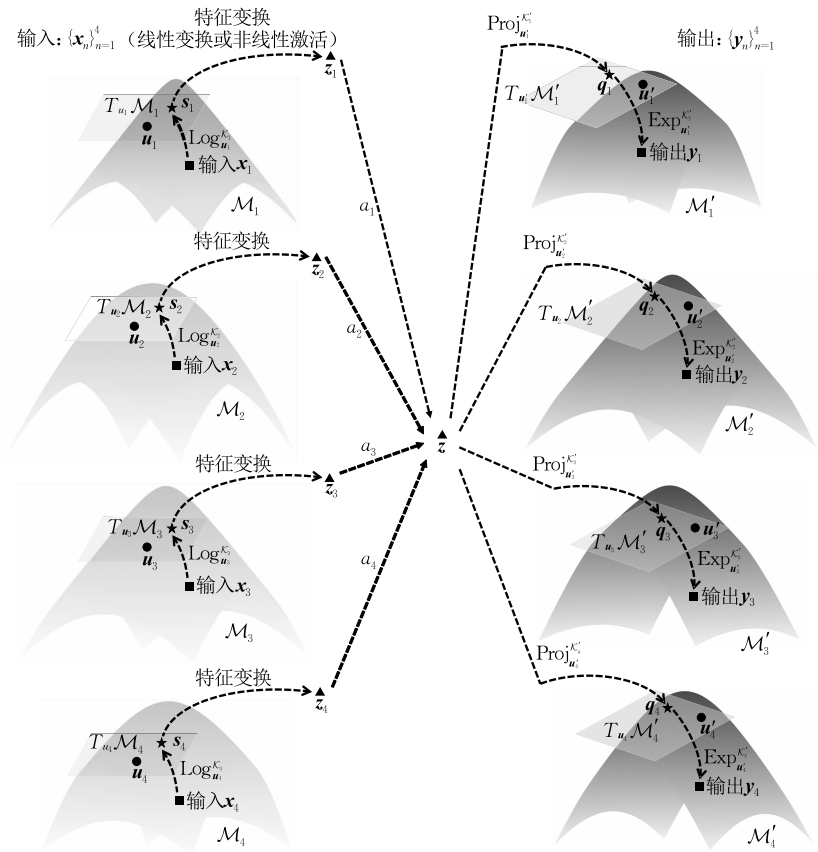


图 1 多混合曲率神经网络中的数据流示意图

实际上,切空间仅仅是混合曲率空间的局部近似,切空间的表示能力将随着切点与特征间距离的增大而降低. 为此,我们设置可学习切点,并通过梯度下降方式更新切点,使得切点可以尽可能地靠近特征. 总之,我们在多混合曲率神经网络的第 i 层中使用两组不同的切空间 $\{T_{\boldsymbol{u}_n}\mathcal{M}_n\}_{n=1}^{r}$ 和 $\{T_{\boldsymbol{u}'_n}\mathcal{M}'_n\}_{n=1}^{r}$. 第一组切空间 $\{T_{\boldsymbol{u}_n}\mathcal{M}_n\}_{n=1}^{r}$ 用于对第 i 层的输入做变换操作;第二组切空间 $\{T_{\boldsymbol{u}'_n}\mathcal{M}'_n\}_{n=1}^{r}$ 用于将结果映射到第 $i+1$ 层的混合曲率空间.

4.2.1 卷积块

本文以依次包含卷积操作、非线性激活操作、批量归一化操作和池化操作的卷积块为例,介绍如何构建混合曲率空间中的卷积块,如图 2 所示. 卷积块的输入是 r 个三维张量 $\{\mathcal{X}_n\in\mathbb{R}^{c\times h\times w}\}_{n=1}^{r}$,分别表示数据在 r 个混合曲率空间 $\mathcal{M}_1,\mathcal{M}_2,\cdots,\mathcal{M}_r$ 中的特征. 张量中 c 表示通道的数量,h 和 w 表示张量的高和宽. 这里将 $\mathcal{X}_n$ 看作是包含了 hw 条位于混合曲率空间 $\mathcal{M}_n$ 中的特征,而混合曲率空间的维度是 c.

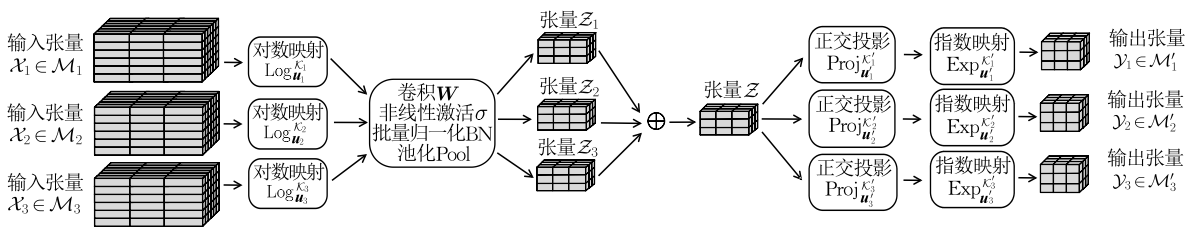


图 2 多混合曲率神经网络中的卷积块示意图

首先使用 r 个对数映射 $\{\mathrm{Logm}_{\boldsymbol{u}_n}^{\mathcal{K}_n}\}_{n=1}^{r}$ 将 r 个三维张量 $\{\mathcal{X}_n\}_{n=1}^{r}$ 中的特征分别映射至对应的切空间 $\{T_{\boldsymbol{u}_n}\mathcal{M}_n\}_{n=1}^{r}$,

$$\mathcal{S}_n=\mathrm{Logm}_{\boldsymbol{u}_n}^{\mathcal{K}_n}(\mathcal{X}_n) \tag{4}$$

$\mathcal{S}_n$是与$\mathcal{X}_n$同样尺寸的三维张量，包含hw个维度为c且位于$T_{\boldsymbol{u}_n}\mathcal{M}_n$中的特征. 在切空间$T_{\boldsymbol{u}_n}\mathcal{M}_n$中对$\mathcal{S}_n$进行卷积、非线性激活、批量归一化和池化操作，

$$\mathcal{Z}_n=\mathrm{Pool}(\mathrm{BN}(\sigma(\boldsymbol{W}\otimes\mathcal{S}_n+\boldsymbol{b}))) \tag{5}$$

其中$\otimes$、σ、BN 和 Pool 分别表示卷积、非线性激活、批量归一化和池化操作，$\boldsymbol{W}$和$\boldsymbol{b}$分别表示卷积操作的权重和偏置. $\mathcal{Z}_n\in\mathbb{R}^{c'\times h'\times w'}$包含了$h'w'$条维度为$c'$的特征，特征位于任意空间. 对于不同混合曲率空间中的特征，使用相同的卷积参数.

此时，张量$\{\mathcal{Z}_n\}_{n=1}^r$中的特征都位于任意空间，可以对此$r$个张量进行加权组合，

$$\mathcal{Z}=\sum_{n=1}^{r}a_n\mathcal{Z}_n \tag{6}$$

其中，a_n表示第n个混合曲率空间的权重. $\mathcal{Z}\in\mathbb{R}^{c'\times h'\times w'}$与$\mathcal{Z}_n$尺寸相同.

卷积块使用r个正交投影操作$\{\mathrm{Proj}_{\boldsymbol{u}_n}^{\mathcal{K}_n}\}_{n=1}^r$将$\mathcal{Z}$中的$h'w'$条特征映射至神经网络下一层的切空间$\{T_{\boldsymbol{u}'_n}\mathcal{M}'_n\}_{n=1}^r$中，

$$\mathcal{Q}_n=\mathrm{Proj}_{\boldsymbol{u}'_n}^{\mathcal{K}'_n}(\mathcal{Z}) \tag{7}$$

其中$\mathcal{Q}_n\in\mathbb{R}^{c'\times h'\times w'}$. 最终使用$r$个指数映射$\{\mathrm{Expm}_{\boldsymbol{u}'_n}^{\mathcal{K}'_n}\}_{n=1}^r$将$\{\mathcal{Q}_n\}_{n=1}^r$映射至神经网络下一层的混合曲率空间$\{\mathcal{M}'_n\}_{n=1}^r$，

$$\mathcal{Y}_n=\mathrm{Expm}_{\boldsymbol{u}'_n}^{\mathcal{K}'_n}(\mathcal{Q}_n) \tag{8}$$

最终获得r个三维张量$\{\mathcal{Y}_n\}_{n=1}^r$作为下一层的输入.

4.2.2　全连接层

全连接层的输入为数据在r个混合曲率空间$\mathcal{M}_1,\mathcal{M}_2,\cdots,\mathcal{M}_r$中的特征$\{\boldsymbol{x}_n\}_{n=1}^r,\boldsymbol{x}_n\in\mathcal{M}_n$. 首先使用$r$个对数映射$\{\mathrm{Logm}_{\boldsymbol{u}_n}^{\mathcal{K}_n}\}_{n=1}^r$将这$r$条特征$\{\boldsymbol{x}_n\}_{n=1}^r$投影到对应的切空间$\{T_{\boldsymbol{u}_n}\mathcal{M}_n\}_{n=1}^r$中，

$$\boldsymbol{s}_n=\mathrm{Logm}_{\boldsymbol{u}_n}^{\mathcal{K}_n}(\boldsymbol{x}_n) \tag{9}$$

在切空间$\{T_{\boldsymbol{u}_n}\mathcal{M}_n\}_{n=1}^r$中对$\{\boldsymbol{s}_n\}_{n=1}^r$应用线性映射和非线性激活操作，

$$\boldsymbol{z}_n=\sigma(\boldsymbol{W}\boldsymbol{s}_n+\boldsymbol{b}) \tag{10}$$

这里$\boldsymbol{W}$与$\boldsymbol{b}$表示全连接层的权重和偏置. 对于不同混合曲率空间中的特征，使用相同的权重和偏置. 与卷积块类似，$\boldsymbol{z}_n$位于任意空间，本文对r条特征$\{\boldsymbol{z}_n\}_{n=1}^r$进行加权组合，

$$\boldsymbol{z}=\sum_{n=1}^{r}a_n\boldsymbol{z}_n \tag{11}$$

通过r个正交投影操作$\{\mathrm{Proj}_{\boldsymbol{u}'_n}^{\mathcal{K}'_n}\}_{n=1}^r$，$\boldsymbol{z}$被映射至神经网络下一层的$r$个切空间$\{T_{\boldsymbol{u}'_n}\mathcal{M}'_n\}_{n=1}^r$，

$$\boldsymbol{q}_n=\mathrm{Proj}_{\boldsymbol{u}'_n}^{\mathcal{K}'_n}(\boldsymbol{z}) \tag{12}$$

最后，使用r个指数映射$\{\mathrm{Expm}_{\boldsymbol{u}'_n}^{\mathcal{K}'_n}\}_{n=1}^r$将$\{\boldsymbol{q}_n\}_{n=1}^r$映射至神经网络下一层的混合曲率空间$\{\mathcal{M}'_n\}_{n=1}^r$，

$$\boldsymbol{y}_n=\mathrm{Expm}_{\boldsymbol{u}'_n}^{\mathcal{K}'_n}(\boldsymbol{q}_n) \tag{13}$$

$\boldsymbol{y}_n\in\mathcal{M}'_n$. 最终获得$r$条特征$\{\boldsymbol{y}_n\}_{n=1}^r$作为网络下一层的输入. 图 3 所示的是全连接层的示意图.

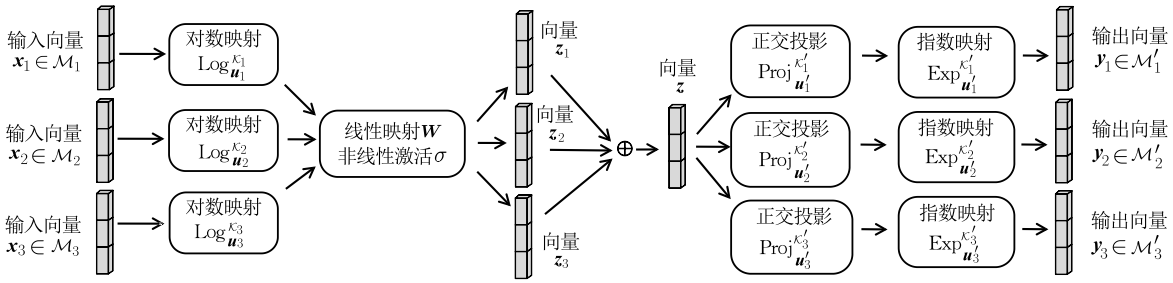


图 3　多混合曲率神经网络中的全连接层示意图

4.2.3　网络架构

在具体实现中，多混合曲率神经网络使用所提出的卷积块和全连接层来代替原欧氏神经网络（例如小样本分类任务中的 ResNet12 网络）中的卷积块和全连接层，同时保持原神经网络的超参数、输出维度和数据变换操作顺序不变. 以 ResNet12 网络为例，该神经网络由四个卷积块和一个全连接层组成，输入图像的大小为 3×84×84. 在第一个卷积块中，输入图像位于原点处切空间中，因此无需对数映射操作. 本文直接遵循欧氏 ResNet12 网络中的数据变换操作顺序（卷积→批量归一化→非线性激活→卷积→批量归一化→非线性激活→卷积→批量归一化→残差链接（卷积→批量归一化→非线性激活））和相应超参数（卷积核大小、卷积步长等），在切空间中对图像执行所需计算，并获得大小为 64×42×42 的张量，64 为通道数量，42 表示长和宽. 接着，使用r个正交投影运算和r个指数映射操作将张量映射到第二个卷积块的r个混合曲率空间中. 第二个卷积块的混合曲率空间的维度为 64. r个对数映射将大小为 64×42×42 的输入张量投影至对应的切空间，按顺序执行所需的操作，加权r个计算结果，最后使用正交投影和指数映射将加权后的张量特征映射到第三个卷积块的r个混合曲率空间，输出的张量大小为 128×21×21. 以此堆叠剩余卷积块和全连接层，获

得多混合曲率神经网络，作为本文方法的骨干网络.

4.3 几何自适应优化

混合曲率空间的几何结构表示为常曲率空间的曲率、维度和数量. 本文将整个多曲率混合神经网络的几何结构表示为 $\boldsymbol{g}=\{\boldsymbol{a},\boldsymbol{\phi}\}$，其中 $\boldsymbol{\phi}=\{\mathcal{K}_1,\cdots,\mathcal{K}_r\}$ 表示所有混合曲率空间的曲率，$\boldsymbol{a}=\{a_1,\cdots,a_r\}$ 表示所有混合曲率空间的权重. 几何自适应优化算法 $\mathcal{A}_\psi$ 以梯度下降方式，对 $\boldsymbol{g}$ 进行更新. ψ 表示几何自适应优化算法的可学习参数，$\mathcal{A}_\psi$ 包括几何初始化生成策略 $\mathcal{I}_{\psi_1}$ 和几何更新策略 $\mathcal{U}_{\psi_2}$，$\psi=\{\psi_1,\psi_2\}$.

具体地，给定任务 $\mathcal{T}_i$，几何初始化生成策略 $\mathcal{I}_{\psi_1}$ 基于公共几何初始化 $\boldsymbol{g}$ 生成任务特定的几何初始化 $\boldsymbol{g}_{i,0}=\mathcal{I}_{\psi_1}(\mathcal{D}_i^s,\boldsymbol{g})$. 几何更新策略 $\mathcal{U}_{\psi_2}$ 采用梯度下降的方式更新 $\boldsymbol{g}_{i,0}$，为梯度下降过程生成自适应的学习率和更新方向，最终获得任务特定的空间几何结构 $\boldsymbol{g}_i'=\mathcal{U}_{\psi_2}(\mathcal{D}_i^s,\boldsymbol{g}_{i,0})$. 同时，本文也采用传统梯度下降方法对多混合曲率神经网络 $\mathcal{N}_{\boldsymbol{g},\boldsymbol{\theta}}$ 的参数 $\boldsymbol{\theta}$ 进行更新. 图 4 所示的是整个自适应过程.

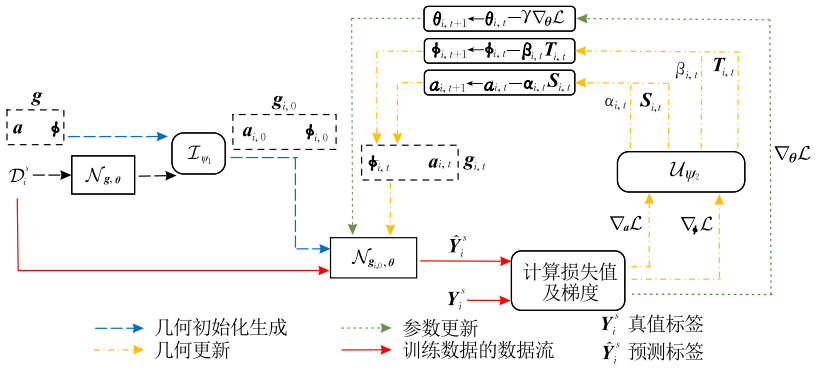


图 4 几何自适应优化的数据流动示意图

4.3.1 几何初始化生成策略

给定新任务 $\mathcal{T}_i$ 及其支持集数据 $\mathcal{D}_i^s$，几何初始化生成策略 $\mathcal{I}_{\psi_1}$ 利用 r 个混合曲率空间的权重 $\boldsymbol{a}$ 和曲率 $\boldsymbol{\phi}$ 的梯度信息，将公共权重初始化 $\boldsymbol{a}$ 和曲率初始化 $\boldsymbol{\phi}$ 转换为任务特定的几何初始化 $\boldsymbol{g}_{i,0}$，包括权重初始化 $\boldsymbol{a}_{i,0}$ 和曲率初始化 $\boldsymbol{\phi}_{i,0}$.

$\mathcal{I}_{\psi_1}$ 计算带有公共初始化的多混合曲率神经网络 $\mathcal{N}_{\boldsymbol{g},\boldsymbol{\theta}}$ 在支持集数据 $\mathcal{D}_i^s$ 上的损失 $\mathcal{L}(\mathcal{N}_{\boldsymbol{g},\boldsymbol{\theta}},\mathcal{D}_i^s)$，并计算 $\mathcal{L}(\mathcal{N}_{\boldsymbol{g},\boldsymbol{\theta}},\mathcal{D}_i^s)$ 关于权重 $\boldsymbol{a}$ 和曲率 $\boldsymbol{\phi}$ 的梯度 $\nabla_a\mathcal{L}(\mathcal{N}_{\boldsymbol{g},\boldsymbol{\theta}},\mathcal{D}_i^s)$，$\nabla_\phi\mathcal{L}(\mathcal{N}_{\boldsymbol{g},\boldsymbol{\theta}},\mathcal{D}_i^s)$. 使用多层感知机对 $\boldsymbol{a}$ 和 $\boldsymbol{\phi}$ 进行调整，

$$\begin{aligned}&\boldsymbol{\gamma}_1=\mathrm{MLP}_1(\boldsymbol{a},\nabla_a\mathcal{L}(\mathcal{N}_{\boldsymbol{g},\boldsymbol{\theta}},\mathcal{D}_i^s)),\\&\boldsymbol{\gamma}_2=\mathrm{MLP}_2(\boldsymbol{\phi},\nabla_\phi\mathcal{L}(\mathcal{N}_{\boldsymbol{g},\boldsymbol{\theta}},\mathcal{D}_i^s)),\\&\boldsymbol{a}_{i,0}=\boldsymbol{\gamma}_1\otimes\boldsymbol{a},\\&\boldsymbol{\phi}_{i,0}=\boldsymbol{\gamma}_2\otimes\boldsymbol{\phi}\end{aligned}\tag{14}$$

其中 $\boldsymbol{\gamma}_1$ 和 $\boldsymbol{\gamma}_2$ 表示缩放因子，$\boldsymbol{\gamma}_1$ 具有与 $\boldsymbol{a}$ 相同的尺寸，$\boldsymbol{\gamma}_2$ 具有与 $\boldsymbol{\phi}$ 相同的尺寸，$\otimes$ 表示矩阵按元素位置的乘法. 几何初始化生成策略 $\mathcal{I}_{\psi_1}$ 的参数 ψ_1 即为多层感知机的参数. MLP_1 和 MLP_2 具体为欧氏空间中的双层全连接层结构，其隐含层维度为 256. 对于生成缩放因子 $\boldsymbol{\gamma}_1$ 的多层感知机，其输入维度和输出维度都与 $\boldsymbol{a}$ 维度相同. 对于生成缩放因子 $\boldsymbol{\gamma}_2$ 的多层感知机，其输入维度和输出维度都与 $\boldsymbol{\phi}$ 维度相同.

4.3.2 几何更新策略

几何更新策略 $\mathcal{U}_{\psi_2}$ 采用梯度下降的方式对 $\boldsymbol{a}_{i,0}$ 和 $\boldsymbol{\phi}_{i,0}$ 进行更新. 考虑到不同的混合曲率空间具有不同的优化轨迹，梯度下降过程中使用单一的学习率可能导致次优的优化结果. 此外，当新任务中训练数据有限时，简单地使用梯度作为更新方向可能会导致优化轨迹的震荡.

为了解决上述问题，几何更新策略对 $\boldsymbol{a}_{i,t}$ 和 $\boldsymbol{\phi}_{i,t}$ 生成自适应的学习率和更新方向，

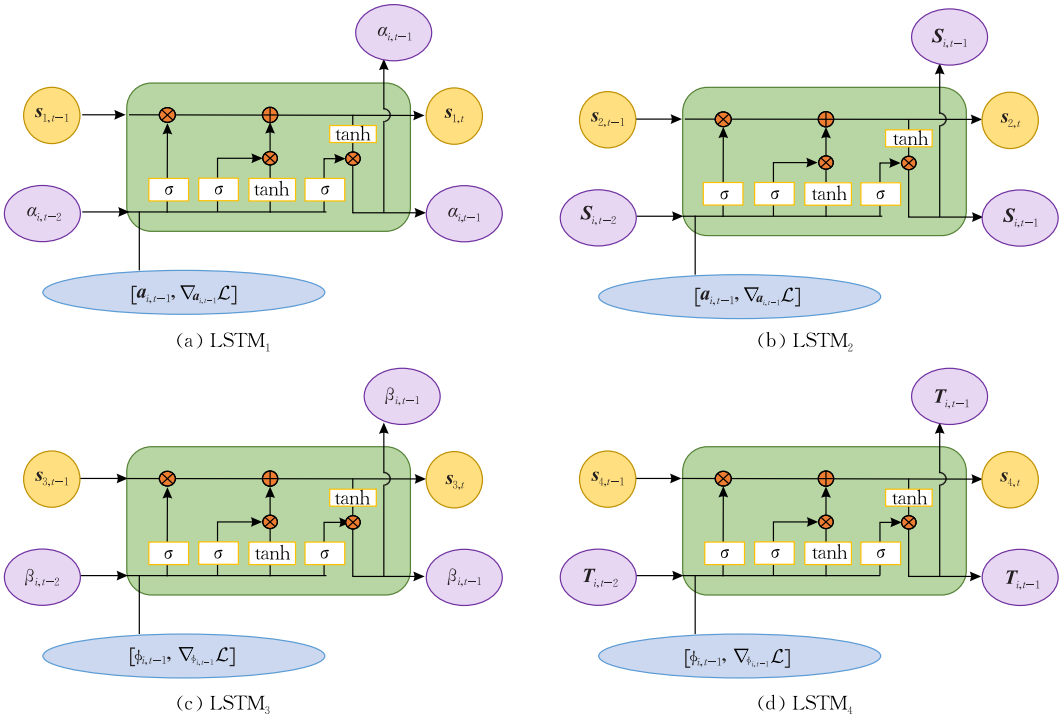
$$\begin{cases}\boldsymbol{a}_{i,t}=\boldsymbol{a}_{i,t-1}-\alpha_{i,t-1}\boldsymbol{S}_{i,t-1},\\\boldsymbol{\phi}_{i,t}=\boldsymbol{\phi}_{i,t-1}-\beta_{i,t-1}\boldsymbol{T}_{i,t-1}\end{cases}\tag{15}$$

其中 $\boldsymbol{a}_{i,t}$ 和 $\boldsymbol{\phi}_{i,t}$ 表示 t 时刻的权重和曲率，$\alpha_{i,t-1}$ 和 $\beta_{i,t-1}$ 分别表示权重和曲率的学习率，$\boldsymbol{S}_{i,t-1}$ 和 $\boldsymbol{T}_{i,t-1}$ 分别表示权重和曲率的更新方向.

本文训练神经网络来自动生成自适应的学习率和更新方向，而不是手动调整学习率和更新方向. 具体地，$\alpha_{i,t-1}$，$\beta_{i,t-1}$，$\boldsymbol{S}_{i,t-1}$ 和 $\boldsymbol{T}_{i,t-1}$ 通过如下方式生成，

$$\begin{cases}\alpha_{i,t-1}=\mathrm{LSTM}_1(\boldsymbol{a}_{i,t-1},\nabla_{\boldsymbol{a}_{i,t-1}}\mathcal{L},\boldsymbol{s}_{1,t-1}),\\\boldsymbol{S}_{i,t-1}=\mathrm{LSTM}_2(\boldsymbol{a}_{i,t-1},\nabla_{\boldsymbol{a}_{i,t-1}}\mathcal{L},\boldsymbol{s}_{2,t-1}),\\\beta_{i,t-1}=\mathrm{LSTM}_3(\boldsymbol{\phi}_{i,t-1},\nabla_{\boldsymbol{\phi}_{i,t-1}}\mathcal{L},\boldsymbol{s}_{3,t-1}),\\\boldsymbol{T}_{i,t-1}=\mathrm{LSTM}_4(\boldsymbol{\phi}_{i,t-1},\nabla_{\boldsymbol{\phi}_{i,t-1}}\mathcal{L},\boldsymbol{s}_{4,t-1})\end{cases}\tag{16}$$

其中 LSTM_1、LSTM_2、LSTM_3 和 LSTM_4 表示 4 个 LSTM 模型，如图 5 所示. $\boldsymbol{s}_{1,t-1}$、$\boldsymbol{s}_{2,t-1}$、$\boldsymbol{s}_{3,t-1}$ 和 $\boldsymbol{s}_{4,t-1}$



(a) LSTM_1　(b) LSTM_2

(c) LSTM_3　(d) LSTM_4

图 5　LSTM 网络结构图

分别表示 4 个 LSTM 模型在 $t-1$ 时刻的优化状态. 相比于多层感知机，LSTM 模型通过考虑之前迭代步的优化状态，获得稳定的优化轨迹. 几何更新策略$\mathcal{U}_{\psi_2}$中的可学习参数 ψ_2即为 4 个 LSTM 模型中的可学习参数.

我们采用传统的梯度下降方式对神经网络的参数初始化 $\boldsymbol{\theta}$ 进行更新，$\boldsymbol{\theta}_{i,t}=\boldsymbol{\theta}_{i,t-1}-\gamma\nabla_{\boldsymbol{\theta}_{i,t-1}}\mathcal{L}$. 经过 T 步更新，获得任务特定的空间几何结构 $\boldsymbol{g}_i'=\boldsymbol{g}_{i,T}$ 和参数 $\boldsymbol{\theta}_i'=\boldsymbol{\theta}_{i,T}$. 当多混合曲率神经网络使用曲率为零的混合曲率空间并在优化过程中使用固定的学习率时，多混合曲率神经网络等价于欧氏空间中的神经网络，并且本文方法也简化为著名的模型无关元学习方法 MAML[1].

4.3.3　训练过程

几何自适应优化算法获得任务特定的多混合曲率神经网络$\mathcal{N}_{\boldsymbol{g}_i',\boldsymbol{\theta}_i'}$后，本文在外层循环中对公共参数初始化 $\boldsymbol{\theta}$、公共几何初始化 $\boldsymbol{g}$、几何自适应优化的参数 $\psi=\{\psi_1,\psi_2\}$进行更新，

$$\begin{cases}\boldsymbol{\theta}\leftarrow\boldsymbol{\theta}-\eta\nabla_{\boldsymbol{\theta}}\sum\limits_{\mathcal{T}_i}\mathcal{L}(\mathcal{N}_{\boldsymbol{g}_{i,T},\boldsymbol{\theta}_{i,T}}\mathcal{D}_i^q),\\ \boldsymbol{g}\leftarrow\boldsymbol{g}-\eta\nabla_{\boldsymbol{g}}\sum\limits_{\mathcal{T}_i}\mathcal{L}(\mathcal{N}_{\boldsymbol{g}_{i,T},\boldsymbol{\theta}_{i,T}},\mathcal{D}_i^q),\\ \psi\leftarrow\psi-\eta\nabla_{\psi}\sum\limits_{\mathcal{T}_i}\mathcal{L}(\mathcal{N}_{\boldsymbol{g}_{i,T},\boldsymbol{\theta}_{i,T}},\mathcal{D}_i^q)\end{cases}\tag{17}$$

η 表示外层循环学习率，训练过程如算法 1 所示.

算法 1.　几何自适应优化训练过程.

输入：任务分布 $p(\mathcal{T})$

输出：更新后的 $\boldsymbol{g},\boldsymbol{\theta},\psi$

1. WHILE {未收敛}
2. 从分布 $p(\mathcal{T})$中采样任务$\mathcal{T}_i$.
3. 构建任务$\mathcal{T}_i$的支持集$\mathcal{D}_i^s$和查询集$\mathcal{D}_i^q$.
4. 使用$\mathcal{D}_i^s$计算多混合曲率神经网络$\mathcal{N}_{\boldsymbol{g},\boldsymbol{\theta}}$的损失$\mathcal{L}(\mathcal{N}_{\boldsymbol{g},\boldsymbol{\theta}},\mathcal{D}_i^s)$和梯度$\nabla_a\mathcal{L}(\mathcal{N}_{\boldsymbol{g},\boldsymbol{\theta}},\mathcal{D}_i^s)$，$\nabla_{\boldsymbol{\Phi}}\mathcal{L}(\mathcal{N}_{\boldsymbol{g},\boldsymbol{\theta}},\mathcal{D}_i^s)$.
5. 根据式(14)计算任务特定的几何初始化 $\boldsymbol{g}_{i,0}=\{\boldsymbol{a}_{i,0},\boldsymbol{\phi}_{i,0}\}$.
6. WHILE ($t\leq T$)
7. 使用使用支持数据$\mathcal{D}_i^s$计算损失$\mathcal{L}(\mathcal{N}_{\boldsymbol{g}_{i,t},\boldsymbol{\theta}_{i,t}},\mathcal{D}_i^s)$，并计算梯度$\nabla_{\boldsymbol{g}_{i,t}}\mathcal{L}(\mathcal{N}_{\boldsymbol{g}_{i,t},\boldsymbol{\theta}_{i,t}},\mathcal{D}_i^s)$，$\nabla_{\boldsymbol{\theta}_{i,t}}\mathcal{L}(\mathcal{N}_{\boldsymbol{g}_{i,t},\boldsymbol{\theta}_{i,t}},\mathcal{D}_i^s)$.
8. 根据式(16)计算几何优化过程中的学习率和更新方向，并根据式(15)通过梯度下降对 $\boldsymbol{g}_{i,t}$更新.
9. 通过梯度下降对参数 $\boldsymbol{\theta}_{i,t}$更新.
10. END WHILE
11. 使用查询集计算任务特定模型$\mathcal{N}_{\boldsymbol{g}_i',\boldsymbol{\theta}_i'}$的损失值$\mathcal{L}(\mathcal{N}_{\boldsymbol{g}_i',\boldsymbol{\theta}_i'},\mathcal{D}_i^q)$.
12. 根据式(17)对 $\boldsymbol{g},\boldsymbol{\theta}$和$\psi$进行更新.
13. END WHILE

4.4　复杂度分析

对于多混合曲率神经网络中的卷积块，其输入三维张量的大小为$\{\mathcal{X}_n\in\mathbb{R}^{c\times h\times w}\}_{n=1}^r$，输出三维张量的大小为$\{\mathcal{Y}_n\in\mathbb{R}^{c'\times h'\times w'}\}_{n=1}^r$，卷积核的大小为 k，则此卷积块的时间复杂度为

$\mathcal{O}(5hwcr+8h'w'c'r+2h'w'c'r+2h'w'k^2cc'r+h'w'c'r)$，其中，对数映射的时间复杂度为$\mathcal{O}(5hwcr)$，指数映射的时间复杂度为$\mathcal{O}(5h'w'c'r)$，正交投影的时间复杂度为$\mathcal{O}(3h'w'c'r)$，卷积与 BN 操作的时间复杂度为$\mathcal{O}(2h'w'k^2cc'r)$，池化与非线性激活操作的时间复杂度为$\mathcal{O}(2h'w'c'r)$，张量加权的时间复杂度为$\mathcal{O}(h'w'c'r)$. 对于多混合曲率神经网络中的全连接层，其输入为$\{\boldsymbol{x}_n\in\mathbb{R}^c\}_{n=1}^r$，输出为$\{\boldsymbol{y}_n\in\mathbb{R}^{c'}\}_{n=1}^r$，其时间复杂度为$\mathcal{O}(5cr+cc'r+10c'r)$，其中对数映射的时间复杂度为$\mathcal{O}(5cr)$，指数映射的时间复杂度为$\mathcal{O}(5c'r)$，正交投影的时间复杂度为$\mathcal{O}(3c'r)$，向量乘法的时间复杂度为$\mathcal{O}(cc'r)$，非线性激活操作的时间复杂度为$\mathcal{O}(c'r)$，向量加权的时间复杂度为$\mathcal{O}(c'r)$.

对于空间复杂度，多混合曲率神经网络卷积块中变量的空间复杂度为$\mathcal{O}(hwcr+6h'w'c'r)$，全连接层中变量的空间复杂度为$\mathcal{O}(c+4c')$.

5 实 验

本文在小样本分类、小样本回归和图像填充任务上评估方法性能. 本文方法表示为 GAML(Geometry-Adaptive Meta-Learning).

5.1 小样本分类

5.1.1 实验设置

小样本分类实验使用两个常用数据集，分别是 mini-ImageNet 数据集[44] 和 tiered-ImageNet 数据集[10]. mini-ImageNet 数据集包含 100 个从原始 ImageNet 数据集中选取的类别，每个类别包含 600 张图像. 本文遵循已有小样本分类方法的数据划分方式[1,10]，从这 100 个类别中分别选取 64、16 和 20 个类别用于训练、验证和测试. tiered-ImageNet 数据集包含 608 个从原始 ImageNet 数据集中选取的类别，共计 779165 张图像，其中 351、97 和 160 个类别分别用于训练、验证和测试. 小样本分类任务通常表示为 l 类 k 样本(k-shot l-way)，即任务包含 l 个类别，每个类别有 k 个样本的分类任务.

实验使用两个骨干网络：ConvNet 和 ResNet12[45]. ConvNet 由 4 层卷积层组成，通道数分别为 3、64、64 和 64. ResNet12 由 4 个卷积块组成，通道数分别为 64、160、320 和 640. 本文分别将 ConvNet 模型 ResNet12 模型扩展至混合曲率空间，保持 ConvNet 和 ResNet12 模型的超参数和输出维度不变.

几何自适应优化过程共包含 $T=5$ 个迭代步. 混合曲率空间的数量 r 和常曲率空间的数量 m 是本文方法的两个重要超参数. 我们通过参数调制的方式确定混合曲率空间的数量 r，我们根据当前任务表示的维度确定常曲率空间的数量 m. 四个卷积块的输出维度为 64、160、320 和 640. m 的选择标准为，能够整除卷积块的输出维度. 为此，我们设置 m 为 16、8、4 和 2. 我们分别尝试了 r 为 2、3、或 4 时方法的性能，并发现当设置 $r=4$ 时，本文方法取得了最优性能. 因此，在实验中，混合曲率空间的数量被设置为 $r=4$. 在这 4 个混合曲率空间中，常曲率空间的数量分别为 16、8、4 和 2. 常曲率空间的曲率被初始化为 -1. 比较方法在 5 类 1 样本和 5 类 5 样本任务上评估性能. 训练阶段包括 20000 个小样本分类任务，测试阶段包括 10000 小样本分类任务. 性能指标为这 10000 个测试任务的平均值准确率.

本文方法分别与欧氏空间中基于模型、基于度量和基于优化的方法，黎曼流形中基于度量和基于优化的方法进行了比较，分别在实验表格中表示为“欧氏-模型”、“欧氏-度量”、“欧氏-优化”、“流形-度量”、“流形-优化”，其中基于优化的方法是 MAML 方法的改进，本文方法属于黎曼流形中基于优化的方法. 虽然使用预训练模型可以提升小样本分类的性能，本实验参照小样本分类的常用准则，所有方法均不使用在额外数据上预训练的模型.

5.1.2 与已有方法比较

mini-ImageNet 和 tiered-ImageNet 数据集上的小样本分类实验结果如表 1 所示. 我们首先与一些著名的欧氏空间中的元学习方法比较，例如 ALFA 和 MeTAL，本文方法取得了更优的性能. 在比较的方法中，ALFA 方法取得了较好的性能. 在 mini-ImageNet 的 1 样本任务和 5 样本任务实验中，本文方法比 ALFA 方法的准确率高大约 2%. LEO 方法使用了更深的骨干模型. LEO 使用 WRN-28-10，本文方法使用 ResNet12. 实验结果显示本文方法更有效. 在 mini-ImageNet 的 1 样本任务和 5 样本任务实验中，本文方法分别比 LEO 的准确率高 2.21% 和 4.16%. tiered-ImageNet 数据集上的实验结果同样显示了本文方法的优越性. 这证明了实际场景中的数据具有非欧结构，为非欧数据构建合适的黎曼几何有助于提升模型性能.

表 1　几何自适应元学习方法在 mini-ImageNet 和 tiered-ImageNet 数据集上的准确率（“欧氏”表示欧氏空间中的元学习方法，“流形”表示黎曼流形中的元学习方法.“优化”、“模型”和“度量”分别指的是基于优化、基于模型和基于度量的元学习方法）（单位：%）

方法	方法种类	骨干模型	mini-ImageNet		tiered-ImageNet	
			1 样本任务	5 样本任务	1 样本任务	5 样本任务
MAML[1]	欧氏-优化	ConvNet	48.70±1.75	63.11±0.91	49.06±0.50	67.48±0.47
FOMAML[1]	欧氏-优化	ConvNet	48.07±1.75	63.15±0.91	50.12±1.82	67.43±1.80
L2F[2]	欧氏-优化	ConvNet	52.10±0.50	69.38±0.46	54.40±0.50	73.34±0.44
TMAML[3]	欧氏-优化	ConvNet	51.77±1.86	65.6±0.93	—	—
iMAML[5]	欧氏-优化	ConvNet	49.30±1.88	63.47±0.90	51.51±1.80	69.92±1.70
CAVIA[6]	欧氏-优化	ConvNet	51.82±0.65	65.85±0.55	—	—
MeTAL[8]	欧氏-优化	ConvNet	52.63±0.37	70.52±0.29	54.34±0.31	70.40±0.21
ALFA[9]	欧氏-优化	ConvNet	52.76±0.52	71.44±0.45	55.06±0.50	73.94±0.43
ProtoNet[23]	欧氏-度量	ConvNet	49.42±0.78	68.20±0.66	53.51±0.89	72.69±0.74
RelationNet[24]	欧氏-度量	ConvNet	50.44±0.82	65.32±0.70	54.48±0.93	71.32±0.78
METAVRF[32]	欧氏-模型	ConvNet	54.20±0.80	67.80±0.70	—	—
Meta-LSTM[34]	欧氏-优化	ConvNet	43.56±0.84	60.60±0.71	—	—
MatchingNet[44]	欧氏-度量	ConvNet	43.56±0.84	55.31±0.73	—	—
MAML++[46]	欧氏-优化	ConvNet	52.15±0.26	68.32±0.44	—	—
HyperProto[12]	流形-优化	ConvNet	54.43±0.20	72.67±0.15	—	—
CurAML[21]	流形-优化	ConvNet	54.66±0.55	72.90±0.50	57.13±0.48	75.70±0.41
GAML(Ours)	流形-优化	ConvNet	**55.24±0.45**	**73.34±0.58**	**59.12±0.86**	**75.96±0.62**
MAML[1]	欧氏-优化	ResNet12	51.03±0.50	68.26±0.47	58.58±0.49	71.24±0.43
L2F[2]	欧氏-优化	ResNet12	57.48±0.49	74.68±0.43	63.94±0.48	77.61±0.41
MeTAL[8]	欧氏-优化	ResNet12	59.64±0.38	76.20±0.19	63.89±0.43	80.14±0.40
ALFA[9]	欧氏-优化	ResNet12	60.05±0.49	77.42±0.42	64.43±0.49	81.77±0.39
ProtoNet[23]	欧氏-度量	ResNet12	56.62±0.45	74.28±0.20	53.51±0.89	72.69±0.74
RelationNet[24]	欧氏-度量	ResNet12	—	—	54.48±0.93	71.32±0.78
SNAIL[31]	欧氏-模型	ResNet12	55.71±0.99	68.88±0.92	—	—
MetaFun[30]	欧氏-模型	ResNet12	—	—	67.27±0.20	83.28±0.12
CAML[47]	欧氏-优化	ResNet12	59.23±0.99	72.35±0.71		
DSN[48]	欧氏-度量	ResNet12	—	—	66.22±0.75	82.79±0.48
MetaOptNet[49]	欧氏-度量	ResNet12	—	—	65.99±0.72	81.56±0.53
LEO[50]	欧氏-优化	WRN-28-10	61.76±0.08	77.59±0.12	66.33±0.05	81.44±0.09
HyperProto[12]	流形-度量	ResNet18	59.47±0.20	76.84±0.14	—	—
HK[16]	流形-度量	ResNet18	61.04±0.21	77.01±0.15	—	—
CurAML[21]	流形-优化	ResNet12	63.13±0.41	81.04±0.39	68.46±0.56	83.84±0.40
GAML(Ours)	流形-优化	ResNet12	**63.97±0.41**	**81.75±0.39**	**69.06±0.56**	**84.19±0.40**

此外，本文方法还与 CurAML[21]、HyperProto[12]和 HK[16]这三个黎曼流形中的元学习方法进行比较.这三个方法同样使用曲率空间建模数据，但它们使用固定的黎曼几何，只能匹配特定的非欧结构.本文方法的空间几何结构具备自适应于不同数据非欧结构的能力.实验结果显示，本文方法的性能优于上述三个方法.在 mini-ImageNet 数据集上，HyperProto 和 HK 方法使用拟合能力更强、模型结构更深的 ResNet18 骨干网络，而我们使用 ResNet12.相比于二者，在 1 训练样本的设定下，本文方法有超过 2% 的准确率提升；在 5 训练样本的设定下，本文方法有超过 3% 的准确率提升.相比于 CurAML 方法，本文方法在 mini-ImageNet 和 tiered-ImageNet 数据集上同样取得了性能提升，显示了自适应空间几何结构的优势.

5.2　小样本回归

5.2.1　实验设置

k 训练样本的正弦函数拟合任务定义为，给定正弦曲线上的 k 个点，训练神经网络拟合这条正弦曲线.遵循已有方法的实验设置[1,9]，小样本回归实验的目标是拟合具有不同振幅、频率和相位的正弦曲线.振幅、频率和相位分别从$[0.1,5.0]$、$[0.8,1.2]$和$[0,\pi]$范围中随机采样.实验包括三个不同的设定，分别提供 $k=5$、$k=10$ 和 $k=20$ 个点作为训练数据.方法性能表示为均方误差.常曲率空间的曲率被初始化为 -1.训练阶段包括 240 个轮次，每个轮次包括 500 个小样本回归任务.测试阶段包括 1000 个任务.训练和测试阶段的优化过程均包含 $T=5$ 个迭代步.使用两个神经网络评估方法性能：一个每层具有 40 个神经元的两层神经网络和一个每层具有

80 个神经元的三层神经网络. 神经网络的输入是曲线横坐标,输出是纵坐标预测值. 由于所使用全连接层的输出维度均为 40 或 80,我们设置了 m 为 5、8、10、20,以对维度进行整除. 在对 r 的参数调制过程中,我们发现保留 $m=5$ 和 $m=10$ 可以取得较好的性能,同时资源消耗较少. 因此,在实验中,混合曲率空间的数量被设置为 $r=2$. 在这两个混合曲率空间中,常曲率空间的数量分别为 5 和 10.

本文方法分别与欧氏空间中基于优化的方法 MAML、TR、ALFA、L2F、MeTAL,以及黎曼流形中基于优化的方法 CurAML 进行了比较. 与小样本分类实验类似,小样本回归实验中的方法均不使用预训练模型.

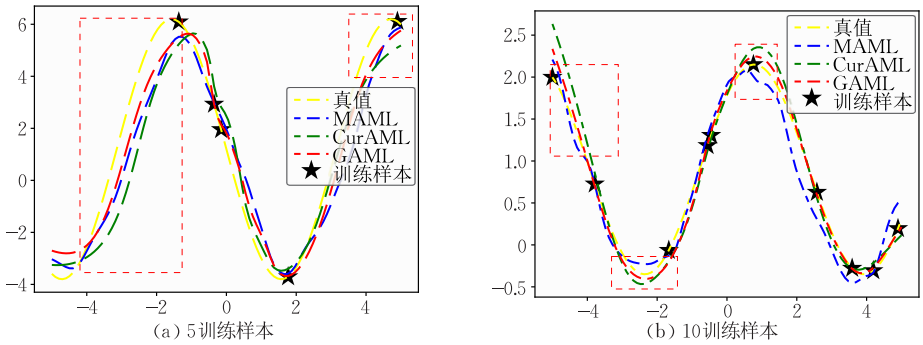
5.2.2 与已有方法比较

正弦函数拟合实验的定量结果如表 2 所示. 与先进的元学习方法 MAML[1]、TR[51]、ALFA[9]、L2F[2]、MeTAL[8]和 CurAML[21]相比,本文方法更准确地拟合了正弦函数曲线,具有更小的均方误差. 即使在训练数据缺乏的情况下,本文方法仍然具有较低的拟合误差. 比较方法 L2F 取得了较好的性能,与之相比,本文方法在使用两层神经网络时,在 5 训练样本、10 训练样本和 20 训练样本的设定下,分别减少了 0.28,0.20 和 0.13 的均方误差. 与流形空间中的元学习方法 CurAML 相比,本文方法在多个实验设定上均取得了更好的性能. 这说明本文提出的几何自适应策略可以更加准确地匹配数据的非欧结构.

表 2 小样本回归实验的均方误差

方法	两层隐含层			三层隐含层		
	5 样本	10 样本	20 样本	5 样本	10 样本	20 样本
MAML[1]	1.24	0.75	0.49	0.84	0.56	0.33
L2F[2]	0.70	0.36	0.16	—	—	—
MeTAL[8]	—	—	—	0.74	0.44	0.21
ALFA[9]	0.92	0.62	0.34	0.70	0.61	0.25
CurAML[21]	0.28	0.09	0.04	0.14	0.06	0.03
TR[51]	1.09	0.66	—	—	—	—
GAML	**0.20**	**0.08**	**0.03**	**0.12**	**0.05**	**0.02**

图 6 中显示了本文方法和 MAML 方法在正弦函数拟合实验中 $k=5$ 和 $k=10$ 两个设定下的定性结果. 实验结果表明,我们方法可以在训练数据较少的情况下,更好地拟合曲线. 在图中,黑色星号是表示正弦曲线的训练数据. 在提供训练数据的区域中,本文方法、MAML、CurAML 方法都能很好地拟合曲线. 在无训练数据的区域,即图中的黑色虚线矩形框区域,本文方法同样可以拟合曲线,具有更小的拟合误差. 例如在 5 训练样本的任务中,当横轴的值位于 −4 至 −2 之间时,我们方法相比于 CurAML 可以更好地拟合曲线.



(a) 5训练样本 (b) 10训练样本

图 6 小样本回归任务定性实验结果

5.3 图像填充实验

5.3.1 实验设置

本文在更具有挑战性的图像填充实验上验证方法性能. 每张图片的填充过程被视为一个独立的任务. 每个任务提供一张像素较少的图像,目标是通过多混合曲率神经网络预测该图像的剩余像素. 图像中给定像素是支持数据,剩余未知像素是查询数据. 图像填充实验使用 CelebA 数据集[52]. 图像填充任务使用由 6 个全连接层组成的神经网络,每层维度为 128. 神经网络的输入是图像中某个像素的坐标,输出是此像素的预测值.

图像填充任务包括 4 个不同的实验设定,以顺序或随机的方式为每张图像提供 10 或 100 个像素作为支持数据. 在实验中,混合曲率空间的数量被设置为 $r=4$,常曲率空间的数量为 2、48、16. 训练阶段包括 5 个轮次,每个轮次包括 10000 个小样本回归任务. 测试阶段包括 1000 个小样本回归任务. 训练和测试阶段的梯度下降过程均包含 $T=5$ 个优化步数.

本文方法分别与欧氏空间中基于模型的方法 CNP,基于优化的方法 MAML 和 CAVIA,以及黎曼流形中基于优化的方法 CurAML 进行了比较. 图像填充实验中的方法均不使用预训练模型.

5.3.2　与已有方法比较

图像填充实验的定量结果如表 3 所示. 观察发现，无论是在随机像素设置还是顺序像素设置中，本文方法比 MAML 方法和 CAVIA 方法取得了更好的性能. 同时，本文方法取得了与 CurAML 可比甚至更优的结果. 这表明设置多个混合曲率空间，可以匹配更多实际数据. 图 7 显示了本文方法与 CurAML 等方法填充图像的结果比较，实验中为每张图片随机提供 10 个像素作为支持数据. 可以观察到，与 CurAML 方法相比，本章方法所填充的图像具有更好的细节. 例如，本文方法生成的图片具有更好的发型细节信息.

表 3　CelebA 数据集上的图像填充任务的均方误差

方法	随机像素		顺序像素	
	10	100	10	100
MAML[1]	0.040	0.017	0.055	0.047
CAVIA[6]	0.037	**0.014**	0.053	0.047
CurAML[21]	0.034	**0.014**	0.052	0.045
CNP[52]	0.039	0.016	0.057	0.047
GAML	**0.032**	**0.014**	**0.051**	**0.042**

注：*MSE* 越低越好.

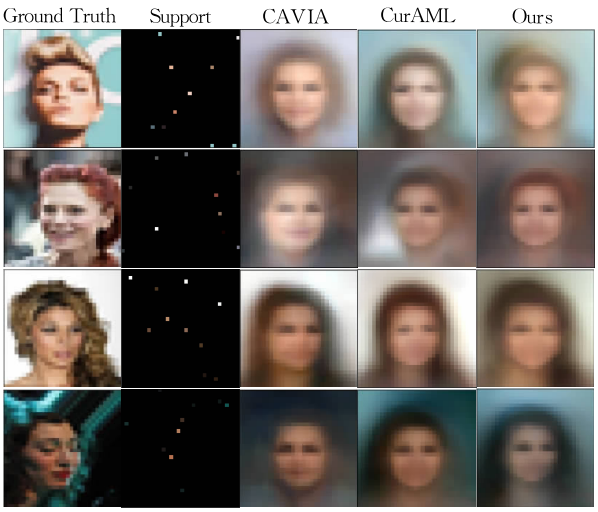


图 7　图像填充任务定性实验结果

5.4　模型拟合能力实验

多混合曲率神经网络的参数数量略大于欧氏神经网络. 多混合曲率神经网络中全连接层和卷积块的权重和偏置参数数量与欧氏神经网络相同，额外参数来自于可学习的切空间切点，但切点所带来额外参数数量较少. 以 ResNet12 为例，欧氏神经网络具有 8029632 个参数，而多混合曲率神经网络具有 8036032 个参数，额外的 6400 个参数是神经网络中的可学习切点.

为了验证方法性能的提升是否主要来自于额外的模型参数，本文设计了 mini-ImageNet 数据集上的模型拟合能力实验. 在此实验中，将多混合曲率神经网络 ResNet12 模型第一个卷积块的输出维度由 64 调整为 62，以此移除神经网络中 7000 多个可学习参数. 此时，多混合曲率神经网络的参数数量略少于欧氏神经网络. 被移除参数的神经网络表示为“GAML, w/o 7000”，具体实验结果如表 4 所示. “GAML, w/o 7000”的性能仅仅略低于 GAML 方法，但仍高于 ALFA 等方法. 这表明，减少 7000 个参数对本文方法影响较小，本文方法的性能提升不是来自于额外的模型参数，而是来自于神经网络的几何自适应能力. 同时，本文方法减少了 7000 个参数后，“GAML, w/o7000” 比 CurAML 方法少了约 6000 个参数. 此时，本文方法性能略低了 CurAML(1 样本任务性能减少了 0.49%，5 样本任务性能减少了 0.26%) 进一步说明了减少参数量对本文方法影响较小.

表 4　模型拟合能力实验　（单位：%）

方法	1 样本任务	5 样本任务
MAML[1]	51.03±0.50	68.26±0.47
L2F[2]	57.48±0.49	74.68±0.43
MeTAL[8]	59.64±0.38	76.20±0.19
ALFA[9]	60.06±0.49	77.42±0.42
CurAML[21]	63.13±0.41	81.04±0.39
CAML[47]	59.23±0.99	72.35±0.71
GAML, w/o 7000	62.64±0.46	80.78±0.37
GAML(Ours)	**63.97±0.41**	**81.75±0.39**

本文通过标准监督学习实验对多混合曲率神经网络的性能做进一步评估. 此实验使用 mini-ImageNet数据集中 64 个类别的训练数据以标准监督学习方式训练神经网络，而不是以双层优化方式训练神经网络. 在测试阶段，多混合曲率神经网络移除了 softmax 分类器，并提取验证集数据 16 个类别数据的特征. 通过计算验证集数据间的距离，对验证集数据进行分类. 此实验使用 ResNet12 骨干网络，测试了将混合曲率空间的数量分别设置为 $r=1$ 时和 $r=4$ 时，神经网络的性能，实验结果见表 5. 与欧氏神经网络相比，使用混合曲率空间可以有效提高监督学习的性能，使用 1 个混合曲率空间将分类准确率提升了 2.33%；使用 4 个混合曲率空间可以进一步提升分类准确率 0.91%. 这验证了多混合曲率神经网络的有效性.

表 5　多混合曲率神经网络模型性能实验（单位：%）

模型	准确率
欧氏神经网络	75.21
多混合曲率神经网络($r=1$)	77.54
多混合曲率神经网络($r=4$)	**78.45**

5.5 消融实验

消融实验用于评估自适应学习率和更新方向的有效性.具体地,本文用“固定 lr”表示在优化过程中为所有混合曲率空间使用固定的学习率;用“生成单一 lr”表示在每一步的梯度下降过程中为所有混合曲率空间生成相同的学习率;使用“梯度”表示在梯度下降过程中直接使用梯度作为更新方向.在 mini-ImageNet 数据集上的实验结果如表 6 所示.实验结果显示,相比于使用固定学习率,为梯度下降过程的每一步生成自适应学习率可以有效提升方法性能(“固定 lr”与“生成单一 lr”相比),如在 1 样本任务上取得了 1.61%的性能提升,在 5 样本任务上取得了 2.21%的性能提升.同时,为不同混合曲率空间生成自适应的学习率可以进一步提升性能,即与“生成单一 lr”相比,GAML 在两个任务上分别取得了 1.53%与 1.43%的提升.此外,对于梯度下降优化过程中的优化方向,与使用梯度作为优化方向(表 6 中的“梯度”)相比,自适应优化方向的方法(GAML)在两个任务上分别带来了 2.08%和 2.17%的提升.因此,通过上述实验可以说明,在梯度下降过程中生成自适应的学习率和更新方向可以显著提升方法性能.

表 6 消融实验结果 (单位:%)

方法	1 样本任务	5 样本任务
固定 lr	60.83±0.41	78.22±0.43
生成单一 lr	62.44±0.40	80.43±0.35
梯度	61.89±0.41	79.58±0.41
GAML(Ours)	**63.97±0.41**	**81.75±0.39**

5.6 可视化实验

5.6.1 优化过程

本文绘制了梯度下降过程中的损失值曲线,将几何自适应优化与手工设计的固定优化方式进行比较.具体地,该实验使用 mini-ImageNet 数据集,在 5 训练样本任务上进行实验,并绘制了使用几何自适应优化和传统梯度下降优化的损失值,如图 8 所示.与手工设计的固定优化方式相比,几何自适应优化具有更快的收敛速度和更优的最优值.

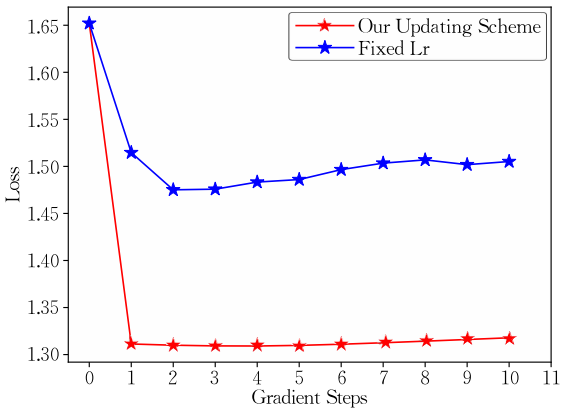


图 8 几何更新策略损失函数值曲线

5.6.2 数据分布

本实验将分类器输入的特征分布可视化,并将本文方法的特征分布与 MAML 方法的特征分布进行比较.该实验从 mini-ImageNet 数据集中采样了 5 训练样本任务,使用骨干网络提取特征,并使用多维尺度变换(Multi-Dimensional Scaling,MDS)降维方法[53]将特征嵌入到二维空间.实验结果如图 9 所示,可以观察到混合曲率空间具有更优的特征分布.在右图的中心区域,欧氏空间中的特征分布杂乱无章,而左图混合曲率空间中的特征在不同类别间具有清晰的边界.这表明使用欧氏空间建模具有非欧结构的数据会导致数据失真问题,而本文方法为数据构建了几何自适应的混合曲率空间,获得了更具有判别力的特征.

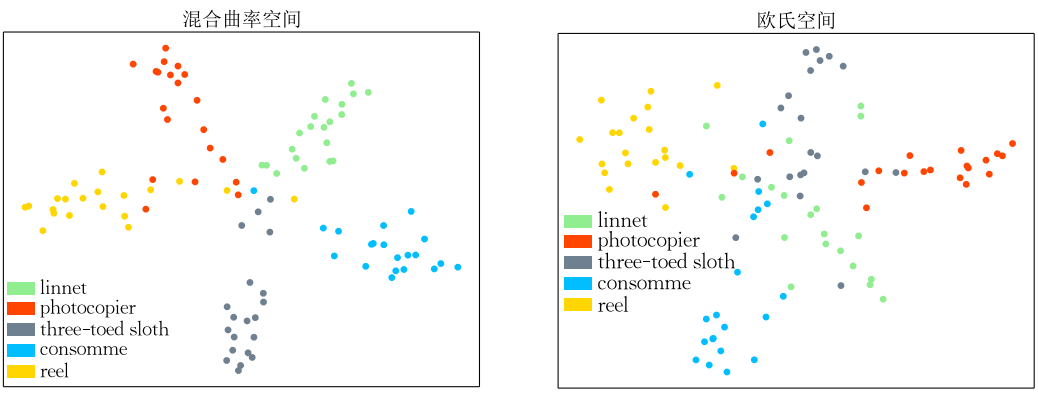


图 9 混合曲率空间与欧氏空间中的特征分布

5.6.3 多混合曲率空间分析

本实验将不同混合曲率空间的曲率、权重和特征进行可视化分析.该实验提取最后一个全连接层各个混合曲率空间的特征表示,并计算表示

在各个混合曲率空间中的距离，最终使用 MDS 算法将特征降为二维. 实验结果如图 10 所示. 该实验为多混合曲率空间的意义提供了一种可能的解释：不同的混合曲率空间捕获数据中不同的信息，例如类别信息. 可以观察到，在 $m=2$ 的混合曲率空间中，黄色类别与其他类别分离较远，可以较好地对黄色类别进行分类；在 $m=4$ 的混合曲率空间中，可以较好地对蓝色类别进行分类；而在 $m=8$ 和 $m=16$ 的混合曲率空间中，可以较好地对绿色、灰色和红色类别进行分类，说明多个混合曲率空间可以分别捕获不同类别的信息，显示了其有效性.

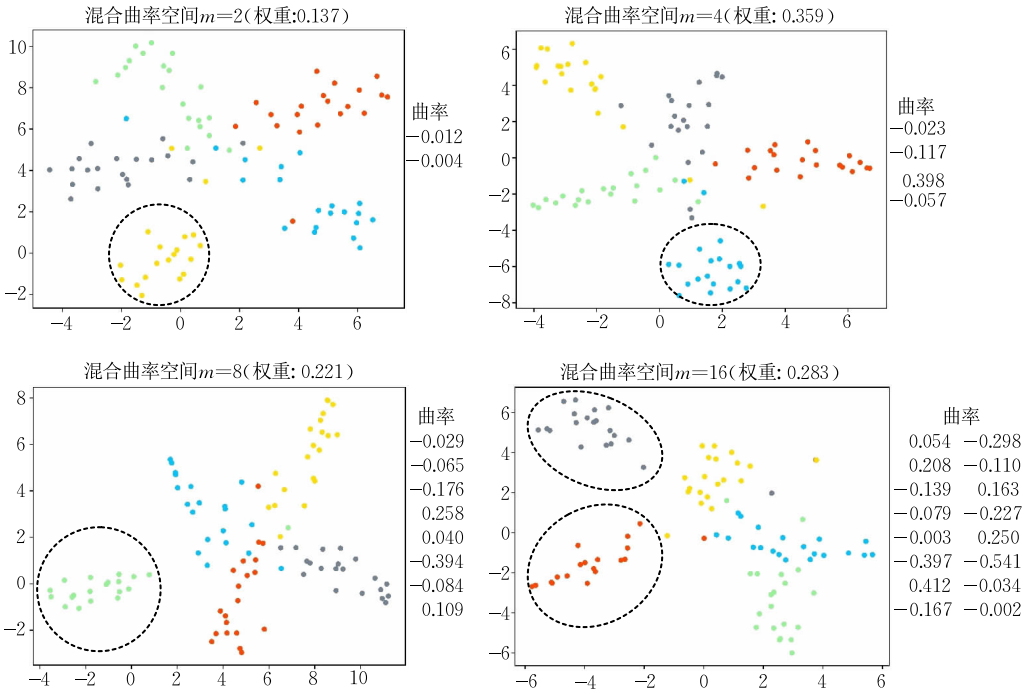


图 10　不同混合曲率空间的特征分布

同时，本文还在图中提供了混合曲率空间所学习到的曲率. 可以看到，负曲率在学习到的曲率中占比较高，说明 mini-ImageNet 数据集中的数据具有较多的层级结构，这与已有研究相一致[12,15].

5.7　效率分析

本实验分析了多混合曲率神经网络的资源消耗. 实验在 mini-ImageNet 数据集的 1-样本任务设定上进行，分别统计了 MAML 方法，CurAML 方法以及本文 GAML 方法在训练和测试过程中的时间消耗. 训练时间测量 500 个小样本任务，测试时间测量 100 个小样本任务. 使用模型为 ConvNet. 所用机器的 CPU 为 Inter(R) Core(TM) i9-10900X 3.7 GHz CPU，显卡为一块 GeForce GTX 3090 GPU，内存大小为 128 GB. 时间消耗如表 7 所示. 此外，本实验还统计了上述方法的内存消耗，如表 8 所示. 实验发现，我们的方法以牺牲计算开销为代价，带来了更好的性能. 这主要原因是多个混合曲率空间增加了资源消耗. 在未来工作中，我们将研究更高效的几何自适应方案，例如通过共享多个混合曲率空间的表示来减少资源消耗.

表 7　时间消耗对比　（单位：s）

方法	1 样本训练	1 样本测试
MAML[1]	154	48
CurAML[21]	538	189
GAML	1836	732

表 8　内存消耗对比　（单位：MB）

方法	内存消耗
MAML[1]	4254
CurAML[21]	6345
GAML	11376

6　结　论

本文提出了几何自适应元学习方法，能够快速适应具有非欧结构的数据. 方法中所搭建的多混和曲率神经网络可以更好地建模真实场景中复杂的数据非欧结构，获得鲁棒的特征表示. 本文方法通过学习任务特定的几何初始化缩短了优化轨迹，几何更新策略自适应地生成学习率和搜索方向，获得更快的收敛速度和更小的最优值. 因此，由几何初始化生

成策略和几何更新策略组成的几何自适应优化可以通过很少的迭代步获得与多样非欧结构相匹配的空间几何结构.本文在小样本分类和小样本回归任务上的实验表明,几何自适应元学习方法取得了比欧氏空间中元学习方法更优的性能.未来工作将研究直接在混合曲率空间中进行几何自适应,无需借助切空间,从而减少指数映射和对数映射带来的计算开销.

参 考 文 献

[1] Finn C, Abbeel P, Levine S. Model-agnostic meta-learning for fast adaptation of deep networks//Proceedings of the International Conference on Machine Learning. Sydney, Australia, 2017: 1126-1135

[2] Baik S, Hong S, Lee K. Learning to forget for meta-learning //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Virtual, 2020: 2376-2384

[3] Jamal M, Qi G, Shah M. Task agnostic meta-learning for few-shot learning//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, USA, 2019: 11719-11727

[4] Li Fan-Zhang, Liu Yang, Wu Peng-Xiang, et al. A survey on recent advances in meta-learning. Chinese Journal of Computers, 2021, 44(2): 422-446(in Chinese)
(李凡长, 刘洋, 吴鹏翔等. 元学习研究综述. 计算机学报, 2021, 44(2): 422-446)

[5] Rajeswaran A, Finn C, Kakade S, et al. Meta-learning with implicit gradients//Advances in Neural Information Processing Systems. Vancouver, Canada, 2019: 113-124

[6] Zintgraf L, Shiarlis K, Kurin V, et al. Fast context adaptation via meta-learning//Proceedings of the International Conference on Machine Learning. Long Beach, USA, 2019: 7693-7702

[7] Li Peng-Fang, Liu Fang, Li Ling-Ling, et al. Meta-feature relearning with embedded label semantics and reweighting for few-shot object detection. Chinese Journal of Computers, 2022, 45(12): 2561-2575(in Chinese)
(李鹏芳, 刘芳, 李玲玲等. 嵌入标签语义的元特征再学习和重加权小样本目标检测. 计算机学报, 2022, 45(12): 2561-2575)

[8] Baik S, Choi J, Kim H, et al. Meta-learning with task-adaptive loss function for few-shot learning//Proceedings of the IEEE/CVF International Conference on Computer Vision. Montreal, Canada, 2021: 9465-9474

[9] Baik S, Choi M, Choi J, et al. Meta-learning with adaptive hyperparameters//Advances in Neural Information Processing Systems. Virtual, 2020: 20755-20765

[10] Jamal M, Wang L, Gong B. A lazy approach to long-horizon gradient-based meta-learning//Proceedings of the IEEE/CVF International Conference on Computer Vision. Montreal, Canada, 2021: 6577-6586

[11] Fan R, Yang C, Vemuri B. Nested hyperbolic spaces for dimensionality reduction and hyperbolic NN design//Proceedings of the IEEE CVF Conference on Computer Vision and Pattern Recognition. New Orleans, USA, 2022: 356-365

[12] Khrulkov V, Mirvakhabova L, Ustinova E, et al. Hyperbolic image embeddings//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle, USA, 2020: 6418-6428

[13] Wang S, Wei X, Santos C, et al. Mixed-curvature multi-relational graph neural network for knowledge graph completion //Proceedings of the Web Conference. Ljubljana, Slovenia, 2021: 1761-1771

[14] Shevkunov K, Prokhorenkova L. Overlapping spaces for compact graph representations//Advances in Neural Information Processing Systems. Virtual, 2021: 11665-11677

[15] Seung H, Lee D. The manifold ways of perception. Science, 2000, 290(5500): 2268-2269

[16] Fang P, Harandi M, Petersson L. Kernel methods in hyperbolic spaces//Proceedings of the IEEE/CVF International Conference on Computer Vision. Montreal, Canada, 2021: 10665-10674

[17] Peng W, Varanka T, Mostafa A. Hyperbolic deep neural networks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, 44(12): 10023-10044

[18] Gu A, Sala F, Gunel B, et al. Learning mixed-curvature representations in product spaces//Proceedings of the International Conference on Learning Representations. New Orleans, USA, 2019: 1-21

[19] Skopek O, Ganea O, Becigneul G. Mixed-curvature variational autoencoders//Proceedings of the International Conference on Learning Representations. Addis Ababa, Ethiopia, 2020: 1-44

[20] Sun L, Zhang Z, Ye J, et al. A self-supervised mixed-curvature graph neural network//Proceedings of the AAAI Conference on Artificial Intelligence. Virtual, 2022: 4146-4155

[21] Gao Z, We Y, Harandi M, et al. Curvature-adaptive meta-learning for fast adaptation to manifold data. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022, doi: 10.1109/TPAMI.2022.3215702

[22] Xu Z, Wen S, Wang J, et al. AMCAD: Adaptive mixed-curvature representation based advertisement retrieval system //Proceedings of the IEEE International Conference on Data Engineering. Kuala Lumpur, Malaysia, 2022: 3439-3452

[23] Snell J, Swersky K, Zemel R. Prototypical networks for few-shot learning//Advances in Neural Information Processing Systems. Long Beach, USA, 2017: 4077-4087

[24] Sung F, Yang Y, Zhang L, et al. Learning to compare: Relation network for few-shot learning//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Salt Lake City, USA, 2018: 1199-1208

[25] Ye H, Hu H, Zhan D, et al. Few-shot learning via embedding adaptation with set-to-set functions//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle, USA, 2020: 8808-8817

[26] Li A, Huang W, Lan X, et al. Boosting few-shot learning with adaptive margin loss//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle, USA, 2020: 12576-12584

[27] Liu J, Song L, Qin Y. Prototype rectification for few-shot learning//Proceedings of the European Conference on Computer Vision. Glasgow, UK, 2020: 741-756

[28] Qi G, Yu H, Lu Z, et al. Transductive few-shot classification on the oblique manifold//Proceedings of the IEEE/CVF International Conference on Computer Vision. Montreal, Canada, 2021: 8412-8422

[29] Santoro A, Bartunov S, Botvinick M, et al. Meta-learning with memory-augmented neural networks//Proceedings of the International Conference on Machine Learning. New York, USA, 2016: 1842-1850

[30] Xu J, Ton J, Kim H, et al. MetaFun: Meta-learning with iterative functional updates//Proceedings of the International Conference on Machine Learning. Virtual, 2020: 10617-10627

[31] Mishra N, Rohaninejad M. Chen X, et al. A simple neural attentive meta-learner//Proceedings of the International Conference on Learning Representations. Vancouver, Canada, 2018: 1-17

[32] Zhen X, Sun H, Du Y, et al. Learning to learn kernels with variational random features//Proceedings of the International Conference on Machine Learning. Virtual, 2020: 11409-11419

[33] Andrychowicz M, Denil M, Gomze S, et al. Learning to learn by gradient descent by gradient descent//Advances in Neural Information Processing Systems. Barcelona, Spain, 2016: 3981-3989

[34] Ravi S, Larochelle H. Optimization as a model for few-shot learning//Proceedings of the International Conference on Learning Representations. Toulon, France, 2017: 1-11

[35] Ganea O, Becigneul G, Hofmann T. Hyperbolic neural networks //Advances in Neural Information Processing Systems. Montréal, Canada, 2018: 5350-5360

[36] Chami I, Gu A, Nguyen D, et al. HoroPCA: Hyperbolic dimensionality reduction via horospherical projections// Proceedings of the International Conference on Machine Learning. Virtual, 2021: 1419-1429

[37] Gao Z, Wu Y, Harandi M, et al. Hyperbolic feature augmentation via distribution estimation and infinite sampling on manifolds//Advances in Neural Information Processing Systems. New Orleans, USA, 2022: 34421-34435

[38] Dai J, Wu Y, Gao Z. et al. A hyperbolic-to-hyperbolic graph convolutional network//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Virtual, 2021: 154-163

[39] Liu Q, Nickel M, Kiela D. Hyperbolic graph neural networks //Advances in Neural Information Processing Systems. Vancouver, Canada, 2019: 8230-8241

[40] Bose A, Smofsky A. Liao R, et al. Latent variable modelling with hyperbolic normalizing flows//Proceedings of the International Conference on Machine Learning. Virtual, 2020: 1045-1055

[41] Huang Z, Gool L. A Riemannian network for SPD matrix learning//Proceedings of the AAAI Conference on Artificial Intelligence. San Francisco, USA, 2017: 2036-2042

[42] Cannon J, Floyd M, Kenyon R, et al. Hyperbolic geometry. Flavors of Geometry, 1997, 31(2): 59-115

[43] Buss S, Fillmore J. Spherical averages and applications to spherical splines and interpolation. ACM Transactions on Graphics, 2001, 20(2): 95-129

[44] Vinyals O, Blundell C, Lillicrap T, et al. Matching networks for one shot learning//Advances in Neural Information Processing Systems. Barcelona, Spain, 2016: 3630-3638

[45] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Las Vegas, USA, 2016: 770-778

[46] Antoniou A, Edwards H, Storkey A. How to train your MAML//Proceedings of the International Conference on Learning Representations. New Orleans, USA, 2019: 1-11

[47] Jiang X, Havaei M, Varno F, et al. Learning to learn with conditional class dependencies//Proceedings of the International Conference on Learning Representations. New Orleans, USA, 2019: 1-11

[48] Simon C, Koniusz P, Nock R, et al. Adaptive subspaces for few-shot learning//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle, USA, 2020: 4135-4144

[49] Lee K, Maji S, Ravichandran A, et al. Meta-learning with differentiable convex optimization//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. Long Beach, USA, 2019: 10657-10665

[50] Rusu A, Rao D, Sygnowski J, et al. Meta-learning with latent embedding optimization//Proceedings of the International Conference on Learning Representations. New Orleans, USA, 2019: 1-17

[51] Collins L, Mokhtari A, Shakkottai S. Task-robust model-agnostic meta-learning//Advances in Neural Information Processing Systems. Virtual, 2020: 18860-18871

[52] Garnelo M, Rosenbaum D, Maddison C, et al. Conditional neural processes//Proceedings of the International Conference on Machine Learning. Stockholm, Sweden, 2018: 1704-1713

[53] Kruskal J. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. Psychometrika, 1964, 29: 1-27

GAO Zhi, Ph. D. His research interests include computer vision, machine learning, and Riemannian geometry.

WU Yu-Wei, Ph. D., tenured associate professor, Ph. D. supervisor. His research interests include computer vision and machine learning.

JIA Yun-De, Ph. D., professor. His research interests include computer vision, computational cognition, and intelligent systems.

Background

Meta-learning is a hot research topic in machine learning. Its goal is to solve problems of deep learning methods that require a lot of training data and long training time. The key idea of meta-learning is learning prior knowledge from seen tasks to achieve fast adaptation to new tasks. Most meta-learning methods use the Euclidean space to represent data, where adaptation is carried out in the Euclidean space. However, much work has shown that various forms of data have non-Euclidean structures. Modeling such data using the Euclidean space will harm the non-Euclidean structures of data, causing inferior generalization to new tasks in meta-learning. Recently, some work focuses on this problem and proposes meta-learning methods for non-Euclidean data by learning non-Euclidean distance measures for new tasks. Although these methods have shown impressive performance, they mainly target classification applications, due to the dependency on distance measures. In addition, they only focus on a specific type of non-Euclidean structure and use a fixed Riemannian geometry to represent data. A fixed Riemannian geometry still cannot handle practical data that has diverse non-Euclidean structures.

In this paper, we propose a geometry-adaptive meta-learning method by using multiple mixed-curvature spaces. We build a multi-mixed-curvature neural network, capable of obtaining discriminative representations for diverse non-Euclidean structures by tuning the geometry. We use learnable tangent spaces to generalize conventional neural network architectures (e. g., convolutional block) to Riemannian manifolds. Compared with fixed tangent spaces that are commonly used, the learnable tangent spaces reduce approximation errors and are more flexible to comply with the Riemannian geometry, providing a new direction to develop Riemannian algorithms. Then, we formulate meta-learning as initializing and optimizing the geometry of mixed-curvature spaces. We assign different weights to mixed-curvature spaces, through which we are allowed to update the geometry in a differentiable way. The geometry of the underlying space adapts to data of new tasks via few optimization steps. In this case, we are able to adapt a whole neural network to new tasks instead of only a distance measure. As such, our method can be applied to a wide set of applications (e. g., regression) based on the network design. The experimental results on multiple benchmarks of the few-shot classification and few-shot regression tasks show that our method can effectively improve the performance of meta-learning.

This work was supported by the National Natural Science Foundation of China under Grants No. 62172041 and No. 62176021, the General Program of Natural Science Foundation of Shenzhen under Grants No. JCYJ20230807142703006, and the Key Research Platform and Program of Guangdong Provincial Department of Education for Ordinary Universities under Grants No. 2023ZDZX1034.