MemKD：面向高效时间序列分类的记忆差异知识蒸馏

Nilushika Udayangani、Kishor Nandakishor 和 Marimuththu Palaniswami电气与电子工程系，
墨尔本大学澳大利亚维多利亚州帕克维尔 3052，hewadehigaha@student.unimelb.edu.au、
nandakishor.desai@unimelb.edu.au 和 palani@unimelb.edu.au

arXiv:2601.04264v1 [cs.LG] 7 Jan 2026

摘要——**深度学习模型，尤其是循环神经网络及其变体（如长短期记忆网络），极大地推动了时间序列数据分析的发展。这些模型能够捕捉时间序列中的复杂序列模式，从而实现实时评估。然而，它们较高的计算复杂度和较大的模型规模给资源受限环境中的部署带来了挑战，例如可穿戴设备和边缘计算平台。知识蒸馏（KD）通过将知识从大型复杂模型（教师模型）迁移到更小、更高效的模型（学生模型），在降低计算需求的同时保持较高性能，为此提供了一种解决方案。当前的知识蒸馏方法最初是为计算机视觉任务设计的，忽视了时间序列模型所特有的时间依赖关系和记忆保持特性。为此，我们提出了一种新型知识蒸馏框架，称为记忆差异知识蒸馏（MemKD）。MemKD利用专门的损失函数，捕捉时间序列数据中各子序列之间教师模型与学生模型的记忆保持差异，确保学生模型能够有效地模仿教师模型的行为。该方法有助于开发紧凑且性能优异的循环神经网络，使其适用于实时时间序列分析任务。大量实验表明，MemKD显著优于最先进的知识蒸馏方法。在保持与教师模型相当性能的同时，它将参数规模和内存使用量降低了约500倍。**

索引词—**知识蒸馏，时间序列分析，循环神经网络**

一、引言

时间序列深度学习模型在众多领域中愈发重要，应用范围从金融预测到工业过程控制。这些模型旨在实现兼顾：既能高效处理复杂序列数据，又能满足轻量化需求，从而部署于边缘设备。循环神经网络（RNN），尤其是长短期记忆网络[1] 和门控循环单元 [2], ，凭借使用隐藏状态对序列数据进行建模的能力，在时间序列分析方面表现出色。然而，在需要持续处理数据并实时预测的实际应用中，RNN 面临诸多挑战。这些局限源于需要通过非线性函数更新大量隐藏状态，而这些函数包含数百万个参数，因此难以满足智能手机和可穿戴设备等边缘设备对低功耗和小体积的要求。

知识蒸馏（KD）将知识从大型教师模型迁移到较小的学生模型，在提升效率的同时保持性能。知识蒸馏由 [3] 提出，并由 [4] 和 [5] 拓展至图像分类领域；研究表明，较小的模型能够媲美甚至超越更深层的模型。尽管适用范围广泛，知识蒸馏在时间序列分析中的研究仍不充分。Ay 等人 [6] 首次研究了将知识蒸馏用于时间序列分类（TSC），采用全卷积网络，并遵循 [4] 中的方法对齐输出概率分布。随后，Ouyang 等人 [7] 提出了用于 TSC 的知识蒸馏框架，通过对齐不同域中的输出分布，在时域和频域提取知识。然而，这些研究主要是将原本为计算机视觉任务开发的知识蒸馏框架加以改造，并未针对时间序列数据的独特特征设计损失函数。与视觉数据不同，时间序列值呈现单向相关性；在分类过程中，短连续子序列通常包含大量判别信息 [8], [9]。因此，将基于图像的方法直接应用于时间序列，可能会无意中破坏这些判别性子序列及其相关性所蕴含的知识。目前专门探索时间序列知识蒸馏的研究有限，尤其是采用循环神经网络教师模型的研究。这些研究包括匹配教师模型与学生模型之间的输出概率分布 [10] ，以及通过对抗学习匹配各自的特征图 [11], [12]。然而，这些方法主要关注不同架构间的知识迁移技术，旨在将循环神经网络的优越性能迁移到卷积神经网络（CNN）等复杂度较低的模型系列。

记忆保持在循环神经网络中的关键作用，为知识蒸馏提出了一条尚未探索的路径：利用教师网络的记忆机制来增强知识迁移。上述研究均未探讨这一问题，这也引出了本文的主要议题：教师网络能否通过向学生网络提供有关其记忆内容的信息来提升学生网络的性能？为此，我们提出了一种新颖的知识蒸馏框架，称为记忆差异知识蒸馏（*MemKD*），如图1所示，该框架基于循环神经网络在输入序列上的记忆保持进行知识迁移。在 MemKD 的核心部分，我们设计了一种新颖的损失函数，促使学生网络模仿教师网络在短连续子序列上的预期记忆保持变化。我们的方法成功地将知识从大型长短期记忆网络-循环神经网络模型蒸馏到小型长短期记忆网络-循环神经网络模型中，在保持与教师模型相当性能的同时，使参数规模和内存使用量减少了约500倍。本文的主要贡献如下：

- MemKD：一种新颖的知识蒸馏框架，可在保持性能的同时，高效压缩用于时间序列分类的复杂长短期记忆网络模型。
- 一种新颖的损失函数，可捕捉并迁移时间序列数据中形状各异的子序列所体现的循环神经网络记忆保持模式。
- 一种特征级知识迁移方法，可在维度表示不同的循环神经网络之间实现无缝的知识流动，克服传统的尺寸匹配限制。
- 在多个数据集上进行全面的实证验证，表明 MemKD 在时间序列任务上的表现显著优于最先进的知识蒸馏方法。

本文其余部分组织如下：第二节回顾相关文献，第三节详细介绍 MemKD 框架，第四节概述实验方法，第五节展示结果，第六节讨论研究发现与未来研究方向。

II. 相关工作

知识蒸馏的根本目标是从教师网络中提取丰富的知识，并将其有效迁移，以指导学生网络的训练。为优化知识质量，研究者提出了各种方法。其中，从软化输出分布中提取知识的 logit级知识蒸馏技术 [4], 仍是应用最广泛的方法。在 logit级知识蒸馏技术的基础上，解耦知识蒸馏（DKD） [13] 通过将 logit级知识解耦为目标类别分布和非目标类别分布来提升性能。除了最终层的 logits，Fitnets [5]提出直接匹配中间层的特征。利用注意力图迁移特征级知识的方法也已有研究 [14], [15]。例如，变分信息蒸馏（VID） [16] 旨在通过变分信息最大化，最大程度地提升学生网络与教师网络之间的互信息；而关系知识蒸馏（RKD） [17] 则侧重于提取输入数据样本之间的关系。这些方法被视为知识蒸馏领域最先进的方法，但它们最初是为计算机视觉任务开发和测试的。因此，其损失函数专门用于捕捉视觉特征，对于时间序列数据可能缺乏直接对应的方法。近期研究才开始通过将这些图像方法改用于时间序列，着手弥补这一差距。例如，文献 [6], [7] 研究了用于时间序列的 logit级知识蒸馏，而 DT2W [18] 则对 Fitnets [5] 进行了扩展，以软动态时间规整距离取代欧几里得距离。

Xu 等人 [11], [12] 在研究不同架构的知识蒸馏方法时，利用对抗学习将 RNN 模型压缩为 CNN。然而，从 RNN 记忆状态中提取知识的可能性尚未得到充分研究。唯一密切相关的研究针对时间序列预测， [19], 其中直接对齐教师模型和学生模型的最后一个隐藏状态。然而，这种方法在两个关键方面存在不足：首先，它主要只关注提升性能，而未涉及模型压缩；其次，它要求教师模型和学生模型采用相同的架构网络。与直接匹配隐藏状态相比，我们提出的方法能够提取丰富的记忆状态知识，使知识得以从大型 RNN 模型高效迁移至紧凑型 RNN 模型，且不受尺寸匹配约束，同时保留关键信息时间信息，并显著缩小模型规模。

III. 记忆差异知识蒸馏

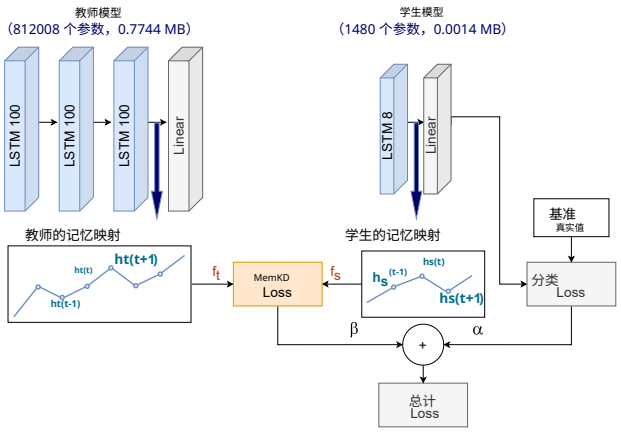

图1：所提出的知识蒸馏框架概览。学生模型通过最小化分类损失来学习目标任务，同时利用中间记忆映射模仿教师网络的记忆保持能力。

A. 问题表述

时间序列可表示为 $\boldsymbol{X} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(T)}\}$，其中 $\boldsymbol{x}^{(i)} \in \mathbb{R}^n$、$i = 1$、$\cdots$、$T$和 $\boldsymbol{X} \in \mathbb{R}^{T \times n}$。时间序列数据集 $\mathcal{D}$包含 M 个样本，是由样本对 $(\boldsymbol{X}_i, \boldsymbol{Y}_i)$ $i = 1$、$\cdots$、M组成的集合，其中 $\boldsymbol{Y}_i \in \mathbb{R}^C$ 是 $\boldsymbol{X}_i$的类别标签。时间序列分类问题涉及训练分类器 $\mathcal{F}$，将 $\boldsymbol{X}_i$ 映射到 $\boldsymbol{Y}_i$。知识蒸馏旨在利用预训练深层教师模型的知识，将浅层学生模型训练为 $\mathcal{F}$，同时最小化蒸馏损失：

$$L_{\mathrm{KD}} = \sum_{\boldsymbol{X}_i} \ell\left(f_t(\boldsymbol{X}_i), f_s(\boldsymbol{X}_i)\right) \tag{1}$$

其中，f_t 和 f_s 分别表示教师模型和学生模型的函数，ℓ 是用于匹配教师模型与学生模型度量的相似度度量。

B. 基于隐藏状态构建知识

给定输入时间序列 $\boldsymbol{X}_i = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(T)}\}$，基本循环神经网络会处理每个输入时间步 $\boldsymbol{x}^{(t)} \in \mathbb{R}^n$，以维护一个随时间变化的记忆状态，即隐藏状态 $\boldsymbol{h}^{(t)} \in \mathbb{R}^m$。在时间步 t，循环神经网络同时利用 $\boldsymbol{x}^{(t)}$和 $\boldsymbol{h}^{(t-1)}$，将隐藏状态 $\boldsymbol{h}^{(t-1)}$更新为 $\boldsymbol{h}^{(t)}$：

$$\boldsymbol{h}^{(t)} = f(\boldsymbol{W}_h \boldsymbol{h}^{(t-1)} + \boldsymbol{W}_i \boldsymbol{x}^{(t)}) \tag{2}$$

其中 $\boldsymbol{W}_h$和 $\boldsymbol{W}_i$是权重矩阵，f是非线性激活函数，用于限制 $\boldsymbol{h}^{(t)}$的取值范围。在

在时间步 T处理完整个序列后，可以在最后一个隐藏状态 $\boldsymbol{h}^{(T)}$上训练一个线性层，以得出用于分类任务的输出概率分布。将式 (2) 中过去的隐藏状态展开后，每个时间步 t 的隐藏状态 $\boldsymbol{h}^{(t)}$ 都可以表示为截至时间步 t 的过去输入值 $\left\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \cdots, \boldsymbol{x}^{(t)}\right\}$ 以及初始隐藏状态值 $\boldsymbol{h}^{(0)}$的函数。循环神经网络中的隐藏状态向量表示网络在每个时间步的记忆，其中封装了截至该时刻已处理的输入序列信息。随着循环神经网络处理输入序列，该向量不断演变，从而捕捉数据中的时间依赖关系和模式。这种对记忆和时间信息的动态表征，使隐藏状态成为知识蒸馏框架潜在的重要知识来源。考虑时间步 $t+z$的隐藏状态 $\boldsymbol{h}^{(t+z)}$， $z \in \mathbb{Z}$。将式 (2) 中各隐藏状态逐一展开至 $\boldsymbol{h}^{(t)}$，可将其写为 $\boldsymbol{h}^{(t+z)}=f(\boldsymbol{h}^{(t)}, \boldsymbol{x}^{(t+1)}, \boldsymbol{x}^{(t+2)}, \cdots, \boldsymbol{x}^{(t+z)})$，其中 f 是一个非线性函数。现在考虑任意两个隐藏状态 $\boldsymbol{h}^{(t+z)}$ 和 $\boldsymbol{h}^{(t)}$ 之间的差异：

$$\Delta \boldsymbol{h}^{t,z}=(\boldsymbol{h}^{(t+z)}-\boldsymbol{h}^{(t)}) \tag{3}$$

这可以看作记忆对输入值子序列 $\left\{\boldsymbol{x}^{(t+1)}, \boldsymbol{x}^{(t+2)}, \cdots, \boldsymbol{x}^{(t+z-1)}, \boldsymbol{x}^{(t+z)}\right\}$的响应，在此记为 $\boldsymbol{X}^{t:(t+z)}$。尽管 $\boldsymbol{h}^{(t+z)}$ 是通过对 $\boldsymbol{h}^{(t)}$ 和 $\boldsymbol{X}^{t:(t+z)}$中的值进行非线性变换计算得到的，但在给定先前历史 $\boldsymbol{h}^{(t)}$ 的情况下，$\Delta \boldsymbol{h}^{t,\ z}$ 可视为由输入值子序列 $\boldsymbol{X}^{t:(t+z)}$ 确定。因此， $\Delta \boldsymbol{h}^{t,\ z}$能够反映模型隐藏状态（记忆）受输入值子序列 $\boldsymbol{X}^{t:(t+z)}$影响的程度。

C. 知识蒸馏目标

我们假设，在时间序列数据中出现某些模式后，模型倾向于记忆随后出现的较小事件或数据模式，而这为提取知识提供了宝贵来源。基于此，我们引入一种新颖的蒸馏损失函数，使用公式（3）中的度量来量化输入子序列对模型记忆的影响。我们的损失函数旨在最小化教师模型与学生模型之间这些度量的差异。我们将公式（1）中的 f_t 和 f_s 定义为：

$$f_t=\left\{\frac{\Delta \boldsymbol{h}_t^{t,z}}{\parallel \boldsymbol{h}_t^{(t)} \parallel}\right\} t, z \in [T-1], t+z \leq T \tag{4}$$

$$f_s=\left\{\frac{\Delta \boldsymbol{h}_s^{t,z}}{\parallel \boldsymbol{h}_s^{(t)} \parallel}\right\} t, z \in [T-1], t+z \leq T \tag{5}$$

其中， $\boldsymbol{h}_t$ 和 $\boldsymbol{h}_s$ 分别表示教师模型和学生模型的隐藏状态向量。为确保损失函数的稳定性，并解决教师模型与学生模型隐藏状态之间的尺度差异，我们在公式（4）和公式（5）中将每个差值除以 $\parallel \boldsymbol{h}^{(t)} \parallel$ 进行归一化。此外，为了衡量 f_t 和f_s之间的差异，我们采用平滑*L1*损失作为记忆差异蒸馏损失：

$$L_{\mathrm{MemKD}}=\sum_{\boldsymbol{X}_i} \ell_{\mathrm{smoothL1}}\left(f_t(\boldsymbol{X}_i), f_s(\boldsymbol{X}_i)\right) \tag{6}$$

表 I：我们的方法与基线方法在 12 个 UCR 数据集上的性能比较。

(a) 与学生基线和 Base-KD 的比较

	Base	Base-KD	我们的方法
平均精确率-召回率曲线下面积	63.42	67.39	**73.66**
平均ROC曲线下面积	87.33	89.73	**92.68**
平均准确率	61.15	64.25	**69.12**
Win	0	1	**11**
Tie	0	0	**0**
Lose	12	11	**1**
平均排名	2.83	2.08	**1.08**

（b）与 Fitnets 的比较

	Fitnet	我们的方法
平均精确率-召回率曲线下面积	68.05	**73.66**
平均 ROC曲线下面积	90.09	**92.68**
平均准确率	65.24	**69.12**
Win	0	**12**
Tie	0	**0**
Lose	12	**0**
平均排名	2	**1**

其中 L_{MemKD} 表示所提出的记忆差异蒸馏损失。平滑L1损失结合了二次项和线性项，相较于L1损失和L2损失各有优势：它比L1损失更平滑，比L2损失对离群值不那么敏感，并有助于防止梯度爆炸 [20]。对于较短的子序列，我们设置 $z=1$并考察$\left\{\Delta \boldsymbol{h}^{t,1}\right\}_{t=1,\ \cdots,\ (T-1)}$，这表示每个输入值导致的记忆状态变化程度。由此构成较短子序列的记忆差异蒸馏损失， $L_{\mathrm{MemKD-Short}}$。对于较长的子序列，我们使用 $z>1$ ，并从 t 和 z中随机选择子集。这使$\left\{\Delta \boldsymbol{h}^{t,z}\right\}$能够捕捉整个时间序列中不同较长子序列的影响，从而形成 $L_{\mathrm{MemKD-Long}}$。我们分别考察了这两个项的作用，发现将它们结合起来能够取得最佳性能。

我们间接计算特征蒸馏损失，无需在教师模型与学生模型的维度之间添加额外的对齐层 [5], [15]。该方法旨在避免直接匹配特征，并更好地保留语义信息。这种方法使特征蒸馏过程更加精简高效，同时不影响效果，也不会增加训练复杂度。

IV. 实验

我们采用三层长短期记忆网络（隐藏层大小为 100）作为教师模型，采用单层长短期记忆网络（隐藏层大小为 8）作为学生模型，实现了 500 倍压缩（见图1）。在评估中，我们从UCR-2015档案中选取了 12 个数据集 [21]，根据时间序列长度 [22], [23] 将其分为短序列（≤150）、中等长度（150-500）和长序列（>500）。该子集包含每种长度类别各 4 个数据集，其中 7 个数据集的输出类别数量较多 [24]。我们保留了原始训练/测试划分，使用 20% 的训练数据作为验证集，并通过将长度标准化为 100 以及进行 Z标准化来预处理数据。

我们使用 ROC曲线下面积（AUC-ROC）、精确率-召回率曲线下面积（AUC-PRC）和准确率评估模型性能，并优先考虑 AUC-PRC，因为它对类别不平衡具有较强的鲁棒性。性能比较采用基于 AUC-PRC 的胜/平/负统计和平均排名。每个数据集上的教师模型均使用五种初始化进行训练，并根据验证集 AUC-PRC 选择表现最佳的模型。学生模型采用多种知识蒸馏方法进行训练：MemKD、Base-KD [4],FitNet [5],关系知识蒸馏 [17], 注意力蒸馏 [14], 解耦知识蒸馏 [13],和 DT2W [18], ，此外还训练了一个不使用知识蒸馏的基线模型（Base）。对于所有使用知识蒸馏的学生模型，我们

表 II：与最先进方法的结果比较。所报告的数值表示五次运行的平均精确率-召回率曲线下面积，但教师模型除外：教师模型同时报告 Top-1 和平均结果（比较时采用后者）。**粗体**数值表示各数据集上学生模型中精确率-召回率曲线下面积的最佳结果。

数据集	教师模型（最大值）	教师模型（平均值）	Base	Base-KD	DKD	Att.	RKD-D	RKD-A	RKD-DA	Fitnet	DT2W	Ours
UWaveGestureLib.All	94.75	91.72	77.15	79.07	80.58	79.31	78.34	78.99	77.51	81.96	81.35	**85.88**
草莓	94.25	87.91	70.68	73.06	71.13	89.09	90.05	91.53	**92.08**	87.1	59.54	90.48
Adiac	70.61	65.79	37.5	45.75	46.36	**49.21**	41.93	43.3	39.94	37.67	43.66	45.07
ItalyPowerDemand	99.24	98.21	94.22	98.63	97.68	97.55	98.37	**99.07**	98.78	97.23	97.94	98.66
yoga	75.88	73.04	57.97	67.41	66.06	70.61	68.36	67.78	69.33	60.1	52.12	**74.01**
Trace	73.84	54.39	64.76	69.25	68.21	61.61	72.71	72.94	72.48	74.11	53.14	**78.76**
ShapesAll	72.34	71.04	29.82	35.34	**44.46**	35.11	30.27	30.98	31.98	35.97	33.55	40.69
瑞典叶片	92.13	89.65	70.75	71.47	72.95	76.39	70.87	72.94	71.11	71.72	73.14	**77.37**
FaceAll	87.16	82.77	46.22	54.38	59.82	58.21	52.68	54.6	57.88	56.08	55.92	**62.26**
MoteStrain	83.89	80.75	82.74	81.87	81.39	82.58	83.02	83.53	82.76	78.16	**84.21**	84.13
无创致命心电图1	86.11	82.29	58.08	57.92	56.96	64.21	61.02	58.4	55.09	65.4	55.01	**69.18**
无创致命心电图2	91.71	86.7	71.2	74.5	66.69	73.26	73.25	73.51	73.41	71.13	72.64	**77.47**
平均精确率-召回率曲线下面积	85.16	80.36	63.42	67.39	67.69	69.76	68.41	68.96	68.53	68.05	63.52	**73.66**
Wins	-	-	0	0	1	1	0	1	1	0	1	**7**
Lose	-	-	12	12	11	11	12	11	11	12	11	**5**
平均排名	-	-	8.83	5.67	5.75	4.58	6.08	4.58	5.42	5.92	6.42	**1.75**
胜场数（使用教师模型）	-	**7**	0	0	0	0	0	1	1	0	1	2
败场数（使用教师模型）	-	**5**	12	12	12	12	12	11	11	12	11	10
平均排名（使用教师模型）	-	3.42	9.67	6.42	6.58	5.33	6.75	5.25	6.08	6.83	7.33	**2.33**

使用蒸馏损失与分类损失（交叉熵损失）的组合对其进行训练，如下所示：

$$L_{\text{train}} = \alpha \times L_{\text{CE}} + \beta \times L_{\text{KD}} \tag{7}$$

其中，α 和 β 分别决定分类损失L_{CE} 和蒸馏损失 L_{KD} 对总训练损失 L_{train}的贡献。在所有实验中，α 固定为 1，而 β 的最优值通过网格搜索选出。每个学生模型均采用五次随机初始化进行训练，评估指标取这五次运行的平均值。所有模型均使用 PyTorch [25] 实现，并采用相同的设置。训练使用 Adam 优化器、批量大小为 32，最多训练 500 轮次，并根据验证损失进行早停。教师模型的初始学习率设为 0.01，学生模型的初始学习率设为 0.1。

五、讨论

我们将 MemKD 的性能与从头开始训练的学生模型（Base）以及传统知识蒸馏（Base-KD）进行了比较。表 Ia 汇总了这两种方法在全部 12 个数据集上的相应结果。显然，我们的方法在全部 12 个数据集上的表现均优于 Base 模型，这表明我们提出的方法是一种高效的知识蒸馏框架。此外，与 Base 模型相比，平均性能指标提升了约 5%–10%。在约 90%（11 个胜出）的数据集上，MemKD 的表现也优于传统知识蒸馏。由于 MemKD 也是一种特征蒸馏方法，我们将其与传统特征蒸馏方法（Fitnets）进行了比较，汇总结果见表 Ib。MemKD 在所有数据集上的表现均优于 Fitnets，平均性能指标提升了 2%–5%。这些结果表明，我们的损失函数能够更充分地利用教师模型特征层中的知识，而非直接提供这些层的提示，同时还免去了额外回归层的需求。表 II 展示了不同知识蒸馏变体在 12 个数据集上的评估结果。我们分别在纳入和不纳入教师模型的情况下，进行了胜/平/负统计。所提出的方法 MemKD 在约 60% 的数据集上始终优于所有其他蒸馏目标。MemKD 的平均排名为 1.75，显著低于其他方法，表明 MemKD 在其余数据集上也具有竞争力。其他方法无一能在至少 10% 的数据集上胜出，排名也均未低于 4。此外，在平均精确率-召回率曲线下面积方面，我们的方法至少比其他方法高出 4%。所有知识蒸馏方法的排名均低于 Base 模型，表明它们都能从知识蒸馏中受益。此外，即使在纳入教师模型的统计中，MemKD 的平均排名仍为最低。就平均排名而言，它是唯一超过教师模型的方法。

六、结论

我们提出了 MemKD，这是一种专为深度时间序列循环网络设计的新型知识蒸馏框架。MemKD 着重研究循环神经网络中的记忆演变，并利用短连续子序列上的记忆保持信息来提升学生模型的性能。MemKD 采用一种能够捕捉记忆保持差异的独特损失函数，有效地将知识从大型长短期记忆网络模型迁移到规模小得多的模型中。在 12 个数据集上的实验表明，MemKD 的表现优于最先进的知识蒸馏方法，同时显著减小模型尺寸和内存使用量，并保持性能。未来的工作将致力于进一步优化 MemKD，并探索其在不同压缩级别下的表现。

致谢

本研究得到墨尔本大学研究计算服务和拍字节级校园倡议的支持。

参考文献

[1] S. Hochreiter，“长短期记忆”，《神经计算》，MIT Press，1997年。[2] K. Cho、B. Van Merri¨enboer、C. Gulcehre、D. Bahdanau、F. Bougares、H. Schwenk和Y. Bengio，“利用循环神经网络编码器-解码器学习用于统计机器翻译的短语表示”，arXiv预印本 arXiv:1406.1078，2014年。[3] J. Ba和R. Caruana，“深度网络真的需要很深吗？”，《神经信息处理系统进展》，第27卷，2014年。[4] G. Hinton、O. Vinyals和J. Dean，“在神经网络中蒸馏知识”，arXiv预印本 arXiv:1503.02531，2015年。[5] A. Romero、N. Ballas、S. E. Kahou、A. Chassang、C. Gatta和Y. Bengio，“Fitnets：用于精简深度网络的提示”，2015年。[6] E. Ay、M. Devanne、J. Weber和G. Forestier，“全卷积网络用于时间序列分类中的知识蒸馏研究”，载于国际神经网络联合会议（IJCNN）。IEEE，2022年，第1–8页。[7] K. Ouyang、Y. Hou、Y. Zhang、C. Ma和S. Zhou，“通过时间域和频域蒸馏实现时间序列分类中的知识迁移”，《应用智能》，第53卷，第2期，第1505–1516页，2023年。[8] P. S. Parvatharaju、R. Doddaiah、T. Hartvigsen和E. A. Rundensteiner，“学习显著性图以解释深度时间序列分类器”，载于ACM信息与知识管理国际会议，2021年，第1406–1415页。[9] L. Ye和E. Keogh，“时间序列形状片段：一种用于数据挖掘的新型基元”，载于ACM知识发现与数据挖掘国际会议，2009年，第947–956页。[10] H. Ma、S. Yang、R. Wu、X. Hao、H. Long和G. He，“基于知识蒸馏的长短期记忆循环神经网络模型加速性能迁移”，《信号、图像与视频处理》，第16卷，第1541–1548页，2022年。[11] Q. Xu、Z. Chen、K. Wu、C. Wang、M. Wu和X. Li，“KDNet-RUL：一种用于压缩机器剩余使用寿命预测深度神经网络的知识蒸馏框架”，《IEEE工业电子学汇刊》，第69卷，第2期，第2022–2032页，2021年。[12] Q. Xu、Z. Chen、M. Ragab、C. Wang、M. Wu和X. Li，“用于时间序列回归任务中深度模型压缩的对比对抗知识蒸馏”，《神经计算》，第485卷，第242–251页，2022年。[13] B. Zhao、Q. Cui、R. Song、Y. Qiu和J. Liang，“解耦知识蒸馏”，载于IEEE/CVF计算机视觉与模式识别会议论文集，2022年，第11 953–11 962页。[14] S. Zagoruyko和N. Komodakis，“更加关注注意力：通过注意力迁移提升卷积神经网络的性能”，arXiv预印本 arXiv:1612.03928，2016年。[15] Z. Huang和N. Wang，“喜欢你所喜欢的：通过神经元选择性迁移进行知识蒸馏”，arXiv预印本 arXiv:1707.01219，2017年。[16] S. Ahn、S. X. Hu、A. Damianou、N. D. Lawrence和Z. Dai，“用于知识迁移的变分信息蒸馏”，载于IEEE/CVF计算机视觉与模式识别会议论文集，2019年，第9163–9171页。[17] W. Park、D. Kim、Y. Lu和M. Cho，“关系知识蒸馏”，载于IEEE/CVF计算机视觉与模式识别会议论文集，2019年，第3967–3976页。[18] Z. Qiao、M. Hu、X. Jiang、P. N. Suganthan和R. Savitha，“通过形状对齐的时间蒸馏实现多变量时间序列的类别增量学习”，载于ICASSP 2023——2023年IEEE声学、语音与信号处理国际会议（ICASSP）。IEEE，2023年，第1–5页。[19] R. Murata、F. Okubo、T. Minematsu、Y. Taniguchi和A. Shimada，“循环神经网络FitNets：通过时间序列知识蒸馏改进学生表现的早期预测”，《教育计算研究》，第61卷，第3期，第639–670页，2023年。[20] R. Girshick，“Fast R-CNN”，载于IEEE计算机视觉国际会议（ICCV）论文集，2015年，第1440–1448页。[21] H. A. Dau、A. Bagnall、K. Kamgar、C.-C. M. Yeh、Y. Zhu、S. Gharghabi、C. A. Ratana mahatana和E. Keogh，“UCR时间序列档案库”，《IEEE/CAA自动化学报》，第6卷，第1293–1305页，2019年。[22] Z. Xiao、X. Xu、H. Zhang和E. Szczerbicki，“一种用于时间序列分类的新型多进程协同架构”，《知识库系统》，第220卷，文章编号106934，2021年。 [在线]。网址：htt
ps://www.sciencedirect.com/science/article/pii/S0950705121001970

[23] H. Xing、Z. Xiao、R. Qu、Z. Zhu 和 B. Zhao，《一种用于多任务时间序列分类的高效联邦蒸馏学习系统》，*IEEE* 仪器与测量汇刊，第 71 卷，12，2022 年，1 页。[24] D. Campos、M. Zhang、B. Yang、T. Kieu、C. Guo 和 C. S. Jensen，《Lightts：通过自适应集成蒸馏实现轻量级时间序列分类》，*ACM* 数据管理会议录，第 1 卷，第 2 期，第 1–27 页，2023 年。[25] A. Paszke、S. Gross、F. Massa、A. Lerer、J. Bradbury、G. Chanan、T. Killeen、Z. Lin、N. Gimelshein、L. Antiga 等，《Pytorch：一种命令式、高性能深度学习库》，神经信息处理系统进展，第 32 卷，2019 年。