文章编号:1672-3961(2024)05-0101-10 DOI:10.6040/j.issn.1672-3961.0.2023.271

基于分解式 Transformer 的联邦长期时间序列预测算法

刘冬兰[1,4*],刘新[1,4],刘家乐[2],赵鹏[3],常英贤[3],王睿[1,4],姚洪磊[1,4],罗昕[2]
(1.国网山东省电力公司电力科学研究院, 山东 济南 250003; 2.山东大学软件学院, 山东 济南 250101; 3.国网山东省电力公司, 山东 济南 250001; 4.山东省智能电网技术创新中心, 山东 济南 250003)

摘要:为解决基于 Transformer 的方法存在计算成本高和无法捕捉时间序列总体趋势的问题,将 Transformer 与季节性趋势分解法相结合,提出基于分解式 Transformer 的联邦长期时间序列预测算法,其中分解方法用于捕捉时间序列的全局概况。在实际场景中,时间序列数据来自多个不同客户端。考虑数据隐私问题,利用联邦学习从多个客户端获得整体最优预测模型,采用基于局部锐度感知最小化的优化器提高全局模型的泛化性。与先进的方法相比,该方法在 4 个基准数据集的多变量和单变量时间序列预测任务中都有改进,在用电负荷(electricity consuming load, ECL)数据集上性能最高可提升 26.9%。试验结果充分表明季节性趋势分解法与局部锐度感知最小化的优化器在长期时间序列预测任务上的有效性。

关键词:隐私保护;联邦学习;长期预测;模型泛化;Transformer
中图分类号:TP183 **文献标志码**:A
引用格式:刘冬兰,刘新,刘家乐,等. 基于分解式 Transformer 的联邦长期时间序列预测算法[J].山东大学学报(工学版),2024,54(5):101-110.
LIU Donglan,LIU Xin,LIU Jiale, et al. Federated long-term time series forecasting algorithm based on decomposed Transformer[J]. Journal of Shandong University (Engineering Science), 2024, 54(5):101-110.

Federated long-term time series forecasting algorithm based on decomposed Transformer

LIU Donglan[1,4*], LIU Xin[1,4], LIU Jiale[2], ZHAO Peng[3], CHANG Yingxian[3], WANG Rui[1,4], YAO Honglei[1,4], LUO Xin[2]
(1. State Grid Shandong Electric Power Research Institute, Jinan 250003, Shandong, China; 2. School of Software, Shandong University, Jinan 250101, Shandong, China; 3. State Grid Shandong Electric Power Company, Jinan 250001, Shandong, China; 4. Shandong Smart Grid Technology Innovation Center, Jinan 250003, Shandong, China)

Abstract: To address the issues of high computational costs and inability to capture the overall trend of time series using Transformer based method, a combined approach of Transformer and seasonal trend decomposition was proposed. A novel federated long-term time series forecasting algorithm based on decomposed Transformer was introduced, where the decomposition method was employed to capture the global overview of time series. In practical scenarios, time series data originated from multiple different clients. Considering data privacy concerns, a federated learning approach was utilized to obtain an overall optimal forecasting model from multiple clients, employing an optimizer based on locally sharpness-aware minimization (SAM) to improve the generalization of the global model. Compared with advanced methods, improvements were observed across multivariate and univariate time series forecasting tasks on four benchmark datasets, with the highest performance enhancement reaching 26.9% on the electricity consuming load (ECL) dataset. Experimental results strongly indicated the effectiveness of seasonal trend decomposition and the SAM optimizer in long-term time series forecasting tasks.

Keywords: privacy protection; federated learning; long-term forecasting; model generalization; Transformer

收稿日期:2023-11-07
基金项目:国网山东省电力公司科技资助项目(520626220018)
第一作者简介:刘冬兰(1987—),女,云南宣威人,高级工程师,硕士,主要研究方向为网络安全、数据安全、隐私计算、区块链等。
E-mail:liudonglan2006@126.com

0 引言

在能源、电力、交通运输和天气等各种应用中[1-3],长期时间序列预测是一个持续性挑战,迫切需要将预测时间延长到遥远的未来,这对长期预测具有重要意义。由于自注意力机制的存在[4],Transformer在对序列数据的长期依赖关系建模方面具有显著优势[5]。由于Transformer对各时间步长的预测是孤立的,预测模型可能无法保持时间序列的整体趋势和特征。为解决这个问题,本研究整合季节性趋势分解法[6-9],将时间序列分析中广泛使用的方法转化为基于Transformer的方法,有效使预测分布与真实分布一致。除了考虑长期时间序列预测性能外,还需要注意与时间序列数据相关的隐私问题。当时间序列数据分布在不同的客户端时,需要在客户端之间不传输数据的情形下聚合客户端模型,得到一个性能良好的全局预测模型。为保护数据隐私,使用联邦学习(federated learning, FL)从多个客户端获得整体最优预测模型[10]。在FL中,过度拟合每个客户端的本地训练数据会降低全局模型的性能。鉴于此,在存在分布偏移问题的情况下,提高全局模型的泛化性是当务之急。提高全局模型的泛化性可以使本地模型的预测性能更接近全局模型的预测性能。近几年,研究人员开发了一种高效算法,称为锐度感知最小化(sharpness-aware minimization, SAM)算法[11-12],利用线性逼近提高模型的收敛效率。将SAM算法应用于FL中,其中每个本地客户端使用相同的扰动训练各自的本地模型,使聚合后的全局模型具有良好的泛化能力。

1 相关工作

1.1 时间序列模型

传统的时间序列预测方法主要是利用基于序列特征的预测技术预测未来趋势。例如,循环神经网络(recurrent neural network, RNN)[13-14]、长短期记忆网络(long short-term memory, LSTM)[15]和门控循环单元(gated recurrent unit, GRU)[16]广泛用于预测时间序列。LSTM以RNN为基础,包含3个门结构:遗忘门、输入门和输出门。尽管LSTM在硬件性能方面要求很高,但在长期时间序列数据上的预测准确率很高。与LSTM相比,GRU优化了3个门函数,减少了参数,降低了硬件要求,加快了模型收敛速度,并获得相似的准确度。GRU和LSTM模型都属于特殊的RNN序列。差分整合移动平均自回归(autoregressive integrated moving average, ARIMA)模型在预测时间序列数据方面具有很高的准确性[17],但需要手动确定多个参数,给批量训练带来挑战。Prophet模型也是一种代表性方法,以简单著称,但往往准确率较低[18]。其他一些传统的时间序列预测方法是基于回归的预测方法。

近几年,利用自注意力机制的Transformer在处理自然语言[4,19]、音频[20]和计算机视觉[21-22]等不同领域的序列数据方面表现出显著能力。对数稀疏Transformer模型(logsparse Transformer, LogTrans)将局部卷积集成到Transformer中,引入LogSparse注意力,选择间隔呈指数增长的时间步长[23]。基于Transformer的长序列时间序列预测模型(Transformer-based model for long sequence time-series forecasting, Informer)扩展Transformer,加入基于Kullback-Leibler(KL)散度的ProbSparse注意力[24]。基于自相关分解Transformer的长期序列预测模型(decomposition Transformers with auto-correlation for long-term series forecasting, Autoformer)用自相关块代替规范注意力,实现子序列级别的注意力[6]。

1.2 联邦学习

联邦学习是一种支持众多客户端的分布式训练框架,客户端可以是移动设备、基站或其他本地信息源[25-26]。针对保护隐私的数据分散问题,联邦学习可以在不与数据交互的情况下提高本地客户端的训练效果。最经典的联邦学习方法联邦平均(federated averaging, FedAvg)接收客户端上传到服务器上的模型参数,简单地对参数取平均,将平均参数返回给每个客户端,忽略了客户端之间的异构性[10]。异构网络中的联邦优化方法(federated optimization in heterogeneous networks, FedProx)在服务器端增加了一个近端项,以改善不同客户端之间结构异构造成的工作性能不一致问题,但对分布偏移问题影响不大[27]。基于知识蒸馏技术,面向异构联邦学习的无数据知识蒸馏方法(data-free knowledge distillation for heterogeneous federated learning, FedGen)聚合本地计算的输出构建全局模型,有助于消除每个本地模型对结构一致性的要求[28]。联邦跨模态检索方法(federated cross-modal retrieval, FedCMR)在客户端分别学习多个模态的公共空间,在服务器端联合学习客户端上传的多个公共子空间,客户端根据服务器端聚合的公共子空

间更新本地空间[29],该方法仅将联邦学习应用于检索领域,对分布偏移问题没有太大改善。模型对比联邦学习方法(model-contrastive federated learning, MOON)在模型层面引入对比学习,关键思想是利用模型表征之间的相似性纠正个体的本地训练[30]。基于动态正则化的联邦学习方法(federated learning based on dynamic regularization, FedDyn)以精确最小化为基础,在每轮中每个参与设备动态更新其正则化器,使正则化损失的最优模型与全局经验损失一致[31]。异构联邦学习中的局部与全局知识蒸馏方法(local-global knowledge distillation in heterogeneous federated learning, FedGKD)是一种新的全局知识蒸馏方法,利用从过去的全局模型中学习到的知识减轻客户端漂移问题[32]。FedGKD 得益于集成和知识蒸馏机制,以产生更准确的模型。动量联邦学习是解决分布偏移问题的一种有效方法,通过将全局信息直接纳入本地训练加快收敛速度。动量可以在服务器端、客户端甚至 2 个层面上实现[33-36]。这些算法虽然加快了收敛速度,但可能会导致全局模型收敛在一个急剧的低谷中,导致过拟合[37-39],最终的全局模型可能无法有效满足所有客户端的需求,导致显著偏差。因此,本研究针对每个本地客户端使用相同的扰动训练各自的本地模型,使全局模型具有良好的泛化能力。

2　本研究方法

2.1　符号和问题定义

假设有 Z 个客户端,其中第 Z 个客户端的数据集为 $\boldsymbol{X}^Z$,整个分布式数据集定义为 $\boldsymbol{D}=\boldsymbol{X}^1\cup\boldsymbol{X}^2\cup\cdots\cup\boldsymbol{X}^Z$。如果第 Z 个客户端在 t 时刻有 m 个时间序列 $\boldsymbol{X}^Z(t)=\{\boldsymbol{X}_1^Z(t),\cdots,\boldsymbol{X}_m^Z(t)\}$,以第 Z 个客户端为例,本研究将 $\boldsymbol{X}^Z(t)$ 简写为 $\boldsymbol{X}(t)=\{\boldsymbol{X}_1(t),\cdots,\boldsymbol{X}_m(t)\}$。

通过对每个时间序列应用傅里叶变换,将每个 $\boldsymbol{X}_i(t)$ 变换成向量 $\boldsymbol{a}_i=(a_{i,1}\quad\cdots\quad a_{i,d})^{\mathrm{T}}\in\mathbf{R}^d$。将所有傅里叶变换后的向量组合成时间序列矩阵 $\boldsymbol{A}=(\boldsymbol{a}_1\quad\cdots\quad\boldsymbol{a}_m)^{\mathrm{T}}\in\mathbf{R}^{m\times d}$,其中每一行对应不同的时间序列,每一列对应一个不同的傅里叶分量。虽然使用所有傅里叶分量可以有效保存时间序列中的历史信息,但可能会导致对历史数据过拟合,进而导致对将来信号的预测不够准确。鉴于此,选择傅里叶分量的一个子集非常重要,该子集应当足够小,以避免过拟合,同时可以保留大部分历史信息。以均匀随机的方式从 d 个傅里叶分量中选择 s 个分量,作为傅里叶变换后时间序列矩阵 $\boldsymbol{A}$ 进行奇异值分解后的对角矩阵的行数,其中 $s<d$。定义 $i_1<i_2<\cdots<i_s$ 为随机选取的分量,构造一个矩阵 $\boldsymbol{S}\in\{0,1\}^{s\times d}$,其中 $i=i_k$ 时,矩阵 $\boldsymbol{S}$ 中的元素 $S_{i,k}=1$,否则 $S_{i,k}=0$,得到多元时间序列 $\boldsymbol{A}'=\boldsymbol{A}\boldsymbol{S}^{\mathrm{T}}\in\mathbf{R}^{m\times s}$。将时间序列矩阵 $\boldsymbol{A}$ 的每个列向量投影到由优化后的 $\boldsymbol{A}'$ 中的列向量构成的子空间中。此时,投影后的矩阵 $P_{\boldsymbol{A}'}(\boldsymbol{A})$ 与原矩阵 $\boldsymbol{A}$ 之间的误差极小,其中 $P_{\boldsymbol{A}'}(\cdot)$ 为投影运算符。这说明,即使傅里叶分量是随机选择的,在适宜条件下,$\boldsymbol{A}'$仍然可以有效保留 $\boldsymbol{A}$ 的大部分信息。

长时间序列预测可以看作一个序列对序列的问题,本研究将输入序列的长度表示为 I,输出序列的长度表示为 O,序列的隐藏状态表示为 D,编码器的输入是一个维度为 $I\times D$ 的矩阵,解码器的输入是一个维度为 $(I/2+O)\times D$ 的矩阵。

2.2　总体结构

受季节性趋势分解法的启发,本研究将 Transformer 更改为一个包含混合专家分解模块的深度分解结构。

编码器采用多层结构 $\boldsymbol{X}_{\mathrm{en}}^l=\mathrm{Encoder}(\boldsymbol{X}_{\mathrm{en}}^{l-1})$,其中,$l$ 为编码层数,$l\in\{1,\cdots,N\}$;Encoder$(\cdot)$计算公式为:

$$\begin{cases}\boldsymbol{S}_{\mathrm{en}}^{l,1}=\mathrm{MED}(\boldsymbol{X}_{\mathrm{en}}^{l-1}),\\ \boldsymbol{S}_{\mathrm{en}}^{l,2}=\mathrm{MED}(\mathrm{FeedForward}(\boldsymbol{S}_{\mathrm{en}}^{l,1})+\boldsymbol{S}_{\mathrm{en}}^{l,1}),\\ \boldsymbol{X}_{\mathrm{en}}^{l}=\boldsymbol{S}_{\mathrm{en}}^{l,2},\end{cases}\tag{1}$$

式中,$\boldsymbol{S}_{\mathrm{en}}^{l,i}(i\in\{1,2\})$为第 l 层第 i 个分解块后的季节项,MED 为混合专家分解模块,FeedForward 为前馈神经网络。初始输入为$\boldsymbol{X}_{\mathrm{en}}^0\in\mathbf{R}^{I\times D}$,表示嵌入的历史序列。解码器采用多层结构 $\boldsymbol{X}_{\mathrm{de}}^g=\mathrm{Decoder}(\boldsymbol{X}_{\mathrm{de}}^{g-1},\boldsymbol{T}_{\mathrm{de}}^{g-1})$,$\boldsymbol{T}_{\mathrm{de}}^g=\mathrm{Decoder}(\boldsymbol{X}_{\mathrm{de}}^{g-1},\boldsymbol{T}_{\mathrm{de}}^{g-1})$,其中,$g$ 为解码层数,$g\in\{1,\cdots,M\}$;Decoder$(\cdot)$计算公式为:

$$\begin{cases}\boldsymbol{S}_{\mathrm{de}}^{g,1}=\mathrm{MED}(\boldsymbol{X}_{\mathrm{de}}^{g-1})\\ \boldsymbol{T}_{\mathrm{de}}^{g,1}=\mathrm{MED}(\boldsymbol{X}_{\mathrm{de}}^{g-1})\\ \boldsymbol{S}_{\mathrm{de}}^{g,2}=\mathrm{MED}(\boldsymbol{S}_{\mathrm{de}}^{g,1})\\ \boldsymbol{T}_{\mathrm{de}}^{g,2}=\mathrm{MED}(\boldsymbol{S}_{\mathrm{de}}^{g,1})\\ \boldsymbol{S}_{\mathrm{de}}^{g,3}=\mathrm{MED}(\mathrm{FeedForward}(\boldsymbol{S}_{\mathrm{de}}^{g,2})+\boldsymbol{S}_{\mathrm{de}}^{g,2})\\ \boldsymbol{T}_{\mathrm{de}}^{g,3}=\mathrm{MED}(\mathrm{FeedForward}(\boldsymbol{S}_{\mathrm{de}}^{g,2})+\boldsymbol{S}_{\mathrm{de}}^{g,2})\\ \boldsymbol{X}_{\mathrm{de}}^{g}=\boldsymbol{S}_{\mathrm{de}}^{g,3}\\ \boldsymbol{T}_{\mathrm{de}}^{g}=\boldsymbol{T}_{\mathrm{de}}^{g-1}+\boldsymbol{W}_{g,1}\boldsymbol{T}_{\mathrm{de}}^{g,1}+\boldsymbol{W}_{g,2}\boldsymbol{T}_{\mathrm{de}}^{g,2}+\boldsymbol{W}_{g,3}\boldsymbol{T}_{\mathrm{de}}^{g,3}\end{cases},\tag{2}$$

式中,$\boldsymbol{S}_{\mathrm{de}}^{g,i}$、$\boldsymbol{T}_{\mathrm{de}}^{g,i}(i\in\{1,2,3\})$分别为第 g 层第 i 个分解块后的季节项和趋势项,$\boldsymbol{W}_{g,i}(i\in\{1,2,3\})$为所提取的第 i 个趋势项 $\boldsymbol{T}_{\mathrm{de}}^{g,i}$的投影。最终的预测是 2 个经过精细分解的分量之和,即 $\boldsymbol{W}_S\boldsymbol{X}_{\mathrm{de}}^M+\boldsymbol{T}_{\mathrm{de}}^M$,其中

$\boldsymbol{W}_S$ 为深度转换后的季节项 $\boldsymbol{X}_{\text{de}}^M$ 在目标维度上的投影。

令 $H_i(\boldsymbol{\omega})=\boldsymbol{L}_i(\boldsymbol{\omega},\boldsymbol{X}^i)$ 为第 i 个客户端的损失函数，其中 $\boldsymbol{\omega}$ 为全局模型，$\boldsymbol{X}^i$ 为第 i 个客户端的本地数据集。经典的联邦学习问题 FedAvg 是通过在每个客户端解决经验风险最小化问题（empirical risk minimization，ERM）$\min\limits_{\omega}\{H(\boldsymbol{\omega})=\dfrac{1}{Z}\sum\limits_{i\in[Z]}\boldsymbol{H}_i(\boldsymbol{\omega})\}$[40]，将最佳全局模型 $\boldsymbol{\omega}$ 拟合到所有样本，其中，$H(\boldsymbol{\omega})$ 为全局模型的损失函数。在联邦学习研究中，处理不同客户端之间异构的本地训练数据集是一个关键挑战，理解联邦学习中本地数据集的非独立同分布（non-independent and identically distributed，Non-IID）特性是基础。现有研究通常假设来自每个客户端的数据样本从常见未知的混合分布转变为局部分布[41-43]。

目前联邦学习中常用的方法是通过随机梯度下降最小化经验风险进行训练，即寻找损失最小的 $\boldsymbol{\omega}$ 有效拟合分布。这可能导致损失曲面收敛于局部最小值[40]，得到的全局模型 $\boldsymbol{\omega}$ 可能会对特定的客户端数据产生偏差，阻碍所有客户端的最优泛化和整体性能。为解决此问题，创建一个更泛化的全局模型，同时考虑平均值和偏差，惠及所有客户端。

目前 SAM 算法的目标是通过向模型添加一个小的扰动 $\boldsymbol{\delta}$ 寻找一个具有低损失的区域，不是寻找像 ERM 这样的单点。由于 FedAvg 中 FL 优化方法的线性特性，通过 SAM 对每个客户端进行扰动损耗训练，可以有效缓解分布偏移的影响，增强全局模型的泛化能力。在局部扰动损失函数 $H_i(\boldsymbol{\omega}+\boldsymbol{\delta}_i)$ 中，对于控制扰动半径一个预定义的常数 ρ，在 $\boldsymbol{\omega}$ 附近使用一阶泰勒展开式，得到线性约束优化

$$\begin{aligned}\boldsymbol{\delta}_i&=\underset{\|\boldsymbol{\delta}_i\|\leqslant\rho}{\operatorname{argmax}}\,H_i(\boldsymbol{\omega}+\boldsymbol{\delta}_i)\approx\\&\operatorname{argmax}\,H_i(\boldsymbol{\omega})+\boldsymbol{\delta}_i^{\mathrm{T}}\nabla H_i(\boldsymbol{\omega})+\boldsymbol{O}(\rho^2)=\\&\rho\operatorname{sign}(\nabla H_i(\boldsymbol{\omega}))\frac{\nabla H_i(\boldsymbol{\omega})}{\|\nabla H_i(\boldsymbol{\omega})\|},\end{aligned}\tag{3}$$

式中，sign(·)为元素方向的符号函数，$\boldsymbol{O}(\rho^2)$ 为一阶泰勒展开式的佩亚诺余项。鉴于此，第 i 个客户端的本地优化目标变为 $\min\limits_{\boldsymbol{\omega}}H_i(\boldsymbol{\omega})=\min\limits_{\tilde{\boldsymbol{\omega}}}f_i(\tilde{\boldsymbol{\omega}})$，其中 $\tilde{\boldsymbol{\omega}}$ 为邻域内损失最大的扰动模型，$\tilde{\boldsymbol{\omega}}=\boldsymbol{\omega}+\rho\dfrac{\nabla H_i(\boldsymbol{\omega})}{\|\nabla H_i(\boldsymbol{\omega})\|}$。本地 SAM 优化器通过迭代 $k=0,\cdots,K-1$ 中的每一轮并进行以下 2 步过程解决压制优化目标中表现最突出的成分问题：

$$\tilde{\boldsymbol{\omega}}_{i,k}^r=\boldsymbol{\omega}_{i,k}^r+\rho\frac{\nabla H_i(\boldsymbol{\omega}_{i,k}^r,\boldsymbol{X}_i^r)}{\left\|\nabla H_i(\boldsymbol{\omega}_{i,k}^r,\boldsymbol{X}_i^r)\right\|},\tag{4}$$

$$\boldsymbol{\omega}_{i,k+1}^r=\boldsymbol{\omega}_{i,k}^r-\eta\nabla H_i(\tilde{\boldsymbol{\omega}}_{i,k}^r,\boldsymbol{X}_i^r),\tag{5}$$

式中 η 为每个客户端本地模型的学习率。从式(4)可以看出，每个客户端的本地训练通过梯度上升近似估计具有固定扰动半径的区域内，在 $\boldsymbol{\omega}_{i,k}^r$ 附近局部损失最大的点为 $\boldsymbol{\omega}_{i,k}^r+\boldsymbol{\delta}_i^r$，根据 $\boldsymbol{\omega}_{i,k}^r+\boldsymbol{\delta}_i^r$ 处的梯度计算在 $\boldsymbol{\omega}_{i,k}^r$ 处的梯度下降程度。根据式(5)可以得到迭代的下一轮中模型的参数值。

2.3 季节性趋势分解的混合专家分解模块

在真实数据中，复杂的季节项通常与趋势项相互耦合，使用固定窗口平均池提取趋势项可能具有挑战性。为解决这一问题，本研究使用一个混合专家分解模块（mixture of experts decomposition，MED），包括各种不同大小的均值滤波器，能够从输入信号中提取多个趋势分量。利用一组依赖于数据的权重组合这些组件并生成最终的季节性趋势

$$\boldsymbol{X}_{\text{trend}}=\operatorname{Softmax}(L(x))\cdot(F(x)),\tag{6}$$

式中，Softmax($L(x)$)为混合提取的趋势权重，$F(\cdot)$为一组均值滤波器。

3 试验

3.1 数据集

本研究在 4 个数据集(5 个案例)上进行广泛试验，涵盖 3 个主流时间序列预测应用，即能源、交通和天气。

变压器温度（electricity transformer temperature，ETT）数据集是电力长调度中的重要指标[5]，包含 2 个子数据集 ETT1 和 ETT2，收集于 2016 年 7 月—2018 年 7 月 2 个站点的 2 个电力变压器，可以将数据集分割为 15 min 级别的ETTm_1、ETTm_2 和1 h级别的ETTh_1、ETTh_2。ETT 由 1 个变压器油温目标值和 6 个负载组成。

用电负荷（electricity consuming load，ECL）数据集包含 321 个客户端在 2012—2014 年每小时的用电量，每一列对应一个客户端[5]。

Traffic 数据集每小时收集一次来自加利福尼亚州运输部的数据[23]，这些数据是旧金山湾区高速公路上不同传感器测量的占用率。

Weather 数据集收集于 2020 年，每 10 min 记录一次，包含气温、湿度等 21 个气象指标[6]。

上述数据集的特征细节如表 1 所示。

表 1　4 个数据集(5 种情况)的特征细节

Table 1　Feature details of four datasets (five cases)

数据集	序列长度	维度	周期/min
$ETTm_2$	69 680	8	15
$ETTh_2$	17 420	8	60
ECL	26 304	322	60
Traffic	17 544	863	60
Weather	52 696	22	10

3.2　方法比较

为验证本研究方法的有效性,将其与几种基线方法(FedProx[27]、MOON[30]、FedDyn[31]、FedGKD[32])进行比较。FedProx 提出一个优化框架,可以处理联邦网络中固有的系统和异构数据,允许跨设备在本地执行不同数量的工作,并依赖于一个近端项帮助稳定方法,可以改善现实异构网络中联邦学习的收敛状态。MOON 在模型层次引入对比学习,核心思想是利用模型表示之间的相似性,修正个体的本地训练。FedDyn 基于精确最小化,在每一轮中,每个参与设备动态更新其正则化器,使正则化损失的最优模型与全局经验损失一致。FedGKD 是一种新的全局知识提取方法,利用从过去的全局模型中学习到的知识缓解客户端漂移问题。FedGKD 利用集成和知识蒸馏机制生成更精确的模型。

3.3　评估指标和试验细节

在试验中,使用 2 个评估指标:均方误差 E_{MS} 和平均绝对误差 E_{MA},$E_{\mathrm{MS}}=\frac{1}{n}\sum_{i=1}^{n}(y-\hat{y})^2$,$E_{\mathrm{MA}}=\frac{1}{n}\sum_{i=1}^{n}|y-\hat{y}|$,其中 y 为模型的预测值,$\hat{y}$ 为输入样本的真实标签。本研究模拟 5 个客户端,将序列数据平均分配到每个客户端。试验中使用自适应矩估计(adaptive moment estimation, ADAM)优化器训练模型[44],批量大小为 32,所有试验设置的学习率为 0.000 1,预定义常数 ρ 设置为 0.05。遵循用于长期序列预测的频率增强分解 Transformer(frequency enhanced decomposed Transformer for long-term series forecasting, FedFormer)试验设置[12],输入长度设置为 96,预测长度分别为 96、192、336、720。本研究分别对单变量时间序列预测和多元时间序列预测任务进行试验。单变量时间序列预测任务主要解决的是输入单变量时间序列,预测单变量未来序列的问题;多元时间序列预测任务则是输入多变量时间序列,预测多变量未来序列,多变量的序列之间存在一定的相互影响关系。相比单变量时间预测,多元时间序列预测更应该考虑建模时间关系的同时对不同变量空间关系进行建模。

3.4　多元时间序列预测结果

经过联邦学习,多元序列预测结果如表 2 所示,E_{MS}、E_{MA}越低,表示性能越好,最佳结果以粗体突出显示。本研究还提供“集中式”的结果,即在一台服务器上将学习到的所有数据进行累加的结果,是联邦学习算法的性能上界。当预测长度为 192 时,$ETTm_2$、$ETTh_2$、ECL、Traffic 和 Weather 的 E_{MS} 损失分别减小 1.5%、0.9%、5.6%、1.2%和 4.5%。随着预测长度增加到 336, $ETTm_2$ 的 E_{MS} 损失减小 1.2%,ECL、Traffic 和 Weather 的 E_{MS} 损失分别减少 1.4%、1.4%和 4.3%。结果表明,SAM 方法有效缓解不同客户端的分布漂移,增强全局模型的泛化能力,提高预测性能。随着预测长度增加,本研究方法始终可以取得最佳结果,说明本研究优化策略适用于长时间序列。

表 2　联邦环境下 4 个数据集(5 种情况)的多变量长时间序列预测结果

Table 2　Multivariate long-term time series forecasting results under federated settings on four datasets (five cases)

数据集	预测长度	集中式		本研究方法		FedGKD		FedDyn		MOON		FedProx	
		E_{MS}	E_{MA}	E_{MS}	E_{MA}	E_{MS}	E_{MA}	E_{MS}	E_{MA}	E_{MS}	E_{MA}	E_{MS}	E_{MA}
$ETTm_2$	96	0.111	0.231	**0.107**	**0.224**	0.108	0.225	0.116	0.232	0.108	0.225	0.123	0.242
	192	0.134	0.251	**0.131**	**0.247**	0.133	0.248	0.140	0.257	0.133	0.248	0.145	0.262
	336	0.166	0.279	**0.166**	**0.274**	0.168	0.276	0.168	0.278	0.168	0.276	0.175	0.286
	720	0.218	0.320	**0.224**	**0.319**	**0.224**	0.320	**0.224**	0.323	0.225	0.320	0.227	0.325
$ETTh_2$	96	0.191	0.302	**0.185**	**0.293**	0.187	**0.293**	0.192	0.298	0.187	**0.293**	0.187	0.294
	192	0.223	0.330	**0.215**	**0.317**	0.217	0.318	0.223	0.323	0.217	**0.317**	0.218	0.319
	336	0.242	0.349	**0.230**	0.332	0.232	**0.331**	0.239	0.338	0.232	0.332	0.236	0.336
	720	0.292	0.385	**0.289**	**0.373**	0.291	**0.373**	0.296	0.375	0.291	0.374	0.292	0.376
ECL	96	0.180	0.294	**0.183**	**0.296**	0.185	0.299	0.292	0.392	0.185	0.300	0.228	0.338
	192	0.191	0.305	**0.198**	**0.310**	0.209	0.321	0.321	0.410	0.209	0.321	0.287	0.380
	336	0.208	0.324	**0.207**	**0.320**	0.210	0.323	0.293	0.388	0.213	0.327	0.267	0.371
	720	0.238	0.347	**0.252**	**0.309**	0.256	0.362	0.372	0.447	0.256	0.362	0.324	0.406

表2(续)

数据集	预测长度	集中式		本研究方法		FedGKD		FedDyn		MOON		FedProx	
		E_{MS}	E_{MA}	E_{MS}	E_{MA}	E_{MS}	E_{MA}	E_{MS}	E_{MA}	E_{MS}	E_{MA}	E_{MS}	E_{MA}
Traffic	96	0.383	0.328	**0.402**	**0.337**	0.412	0.341	0.506	0.422	0.412	0.341	0.464	0.390
	192	0.401	0.335	**0.410**	**0.339**	0.415	0.344	0.531	0.437	0.415	0.343	0.557	0.438
	336	0.415	0.347	**0.420**	**0.343**	0.426	0.351	0.552	0.441	0.426	0.351	0.492	0.409
	720	0.437	0.360	**0.437**	**0.358**	0.439	0.362	0.601	0.474	0.438	0.362	0.540	0.436
Weather	96	0.188	0.268	**0.198**	**0.273**	0.294	0.279	0.226	0.296	0.204	0.279	0.199	0.274
	192	0.254	0.324	**0.244**	**0.304**	0.258	0.331	0.273	0.328	0.255	0.331	0.272	0.324
	336	0.318	0.366	**0.303**	**0.344**	0.328	0.375	0.323	0.355	0.327	0.375	0.316	0.353
	720	0.387	0.405	**0.385**	**0.391**	0.391	0.399	0.400	0.402	0.392	0.398	0.393	0.398

3.5 单变量时间序列预测结果

当预测任务由多变量变为单变量时，再次进行试验，结果如表3所示，最佳结果以粗体突出显示。对于单变量任务，本研究的SAM方法在多个数据集上均实现最佳性能。对于Weather数据集，可以观察到几乎所有方法都执行得非常好，可能是由于天气数据集的固有特征，其中序列的不同部分描述一致的序列信息。

表3 联邦环境下4个数据集(5种情况)的单变量长期时间序列预测结果

Table 3 Univariate long-term time series forecasting results under federated settings on four datasets (five cases)

数据集	预测长度	集中式		本研究方法		FedGKD		FedDyn		MOON		FedProx	
		E_{MS}	E_{MA}	E_{MS}	E_{MA}	E_{MS}	E_{MA}	E_{MS}	E_{MA}	E_{MS}	E_{MA}	E_{MS}	E_{MA}
$ETTm_2$	96	0.100	0.225	**0.100**	**0.222**	**0.100**	**0.222**	0.117	0.251	0.102	0.228	0.150	0.292
	192	0.161	0.297	**0.157**	**0.288**	0.159	0.291	0.184	0.321	0.163	0.297	0.211	0.347
	336	0.234	0.360	**0.231**	**0.355**	**0.231**	**0.355**	0.249	0.374	0.235	0.358	0.277	0.397
	720	0.322	0.436	**0.319**	**0.433**	0.320	**0.433**	0.347	0.455	0.322	0.436	0.351	0.459
$ETTh_2$	96	0.238	0.376	**0.240**	**0.375**	0.243	**0.375**	0.274	0.404	0.245	0.377	0.267	0.400
	192	0.280	0.415	**0.274**	**0.409**	**0.274**	**0.409**	0.299	0.428	0.278	0.411	0.292	0.425
	336	0.300	0.431	**0.294**	**0.423**	0.296	0.426	0.297	0.428	0.295	0.426	0.298	0.427
	720	0.363	0.479	**0.337**	**0.461**	0.339	0.463	0.340	0.464	0.343	0.466	0.341	0.464
ECL	96	0.225	0.342	**0.242**	**0.362**	0.253	0.366	0.358	0.452	0.254	0.367	0.264	0.379
	192	0.260	0.366	**0.282**	**0.381**	0.293	0.390	0.354	0.438	0.284	0.387	0.285	0.389
	336	0.309	0.406	**0.320**	**0.414**	0.329	0.418	0.402	0.470	0.327	0.415	0.346	0.433
	720	0.375	0.455	**0.364**	**0.452**	0.421	0.480	0.526	0.551	0.427	0.484	0.444	0.497
Traffic	96	0.160	0.259	**0.170**	**0.270**	0.202	0.300	0.259	0.376	0.201	0.298	0.214	0.315
	192	0.155	0.258	**0.165**	**0.266**	0.196	0.307	0.261	0.374	0.198	0.310	0.211	0.325
	336	0.164	0.271	**0.169**	**0.276**	0.196	0.312	0.271	0.387	0.194	0.308	0.219	0.333
	720	0.216	0.310	**0.233**	**0.318**	0.259	0.374	0.295	0.387	0.241	0.341	0.260	0.372
Weather	96	0.006	0.057	**0.002**	**0.034**	**0.002**	0.037	**0.002**	0.037	**0.002**	0.038	**0.002**	**0.034**
	192	0.005	0.055	**0.002**	0.038	**0.002**	**0.032**	**0.002**	0.039	**0.002**	0.033	**0.002**	0.036
	336	0.005	0.055	**0.002**	**0.033**	**0.002**	**0.033**	**0.002**	0.039	**0.002**	**0.033**	**0.002**	0.039
	720	0.003	0.044	0.003	**0.040**	0.003	0.041	**0.002**	0.041	0.003	0.043	0.003	0.042

3.6 季节性趋势分解模块效果试验

为验证季节性趋势分解模块和傅里叶变换的有效性，进行进一步的对比试验，结果如表4所示。Reformer使用局部敏感哈希，替换原始点乘方式的注意力机制，与Transformer相比速度更快[45]。Informer扩展了Transformer，加入基于KL散度的ProbSparse注意力[24]。Autoformer采用一种不同的方法，用自相关块代替规范注意力，实现子序列级别的注意力[6]。由表4可以明显看出，在相同的联邦学习条件下，本研究方法始终能够获得更好

的性能。在 $ETTh_2$ 数据集上,短时间序列和长时间序列的 E_{MS} 损失在3.8%~5.4%之间下降。对于 Traffic 数据集,E_{MS} 损失在短时间序列和长时间序列上在5.0%~8.6%之间下降。在 ECL 数据集上,短时间序列和长时间序列 E_{MS} 损失显著降低,分别为 26.8%和24.1%。本研究方法的大幅改进主要归功于季节性趋势分解模块能够更好地将预测分布与真实分布进行对齐。

表 4　不同预测方法的多变量时间序列预测结果

Table 4　Multivariate time series forecasting results for different forecasting methods

数据集	预测长度	本研究方法		Autoformer		Informer		Reformer	
		E_{MS}	E_{MA}	E_{MS}	E_{MA}	E_{MS}	E_{MA}	E_{MS}	E_{MA}
$ETTh_2$	96	**0.185**	**0.293**	0.195	0.301	0.347	0.453	0.464	0.523
	192	**0.215**	**0.317**	0.225	0.326	0.439	0.507	0.683	0.642
	336	**0.230**	**0.332**	0.239	0.338	0.591	0.573	0.928	0.761
	720	**0.289**	**0.373**	0.305	0.381	0.756	0.661	1.117	0.826
ECL	96	**0.183**	**0.296**	0.250	0.354	0.344	0.429	0.309	0.393
	192	**0.198**	**0.310**	0.262	0.365	0.343	0.428	0.352	0.421
	336	**0.207**	**0.320**	0.283	0.382	0.379	0.448	0.395	0.452
	720	**0.252**	**0.309**	0.332	0.418	0.381	0.446	0.473	0.506
Traffic	96	**0.402**	**0.337**	0.435	0.350	0.502	0.385	0.457	0.353
	192	**0.410**	**0.339**	0.446	0.371	0.511	0.386	0.470	0.359
	336	**0.420**	**0.343**	0.442	0.367	0.568	0.417	0.482	0.366
	720	**0.437**	**0.358**	0.478	0.394	0.652	0.469	0.493	0.375

3.7　消融试验

在保证模型结构不变的情况下,本研究通过试验分析联邦学习中的优化策略,结果如表 5 所示,其中“ours w/o SAM”及“ours w/o MED”表示本研究方法的变体,即分别从模型的学习策略中省略 SAM 方法及 MED 模块。由表 5 可以看出,在不同的预测长度下,性能出现不同程度的下降。这一现象表明,SAM 方法对减少来自不同客户端序列数据之间的特征漂移有显著作用,进而验证所提出的 SAM 方法的有效性。在省略 MED 模块后,性能大大下降,充分表明季节性趋势分解模块在预测中的重要性。

表 5　联邦学习环境下的消融试验

Table 5　Ablation study in the setting offederated learning

数据集	预测长度	Ours w/o SAM		Ours w/o MED		本研究方法	
		E_{MS}	E_{MA}	E_{MS}	E_{MA}	E_{MS}	E_{MA}
$ETTh_2$	96	0.187	**0.293**	0.347	0.453	**0.185**	**0.293**
	192	0.217	**0.317**	0.439	0.507	**0.215**	**0.317**
	336	0.231	**0.331**	0.591	0.573	**0.230**	0.332
	720	0.291	**0.373**	0.756	0.661	**0.289**	**0.373**
ECL	96	0.185	0.300	0.344	0.429	**0.183**	**0.296**
	192	0.209	0.321	0.343	0.428	**0.198**	**0.310**
	336	0.213	0.327	0.379	0.448	**0.207**	**0.320**
	720	0.255	0.362	0.381	0.446	**0.252**	**0.309**
Traffic	96	0.413	0.341	0.502	0.385	**0.402**	**0.337**
	192	0.416	0.343	0.511	0.386	**0.410**	**0.339**
	336	0.426	0.351	0.568	0.417	**0.420**	**0.343**
	720	0.438	0.361	0.652	0.469	**0.437**	**0.358**

3.8　超参数优化

对不同选择的 ρ 进行试验,ρ 用于控制扰动损失函数中的扰动半径。适当的 ρ 允许全局模型定位整个数据集损失最小的区域,优化全局模型的整体性能。本研究针对不同数据集下超参数 ρ 的选择进行试验,具体试验结果如图 1~3 所示。从图 1~3 可以看出,当 $\rho=0.05$时,模型的性能最优,整体模型的泛化效果最好。

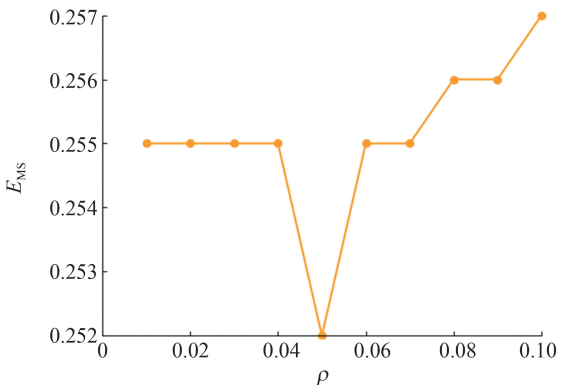

图 1 在 ECL 数据集上微调超参数 ρ 的 E_{MS} 损失变化
Fig.1 Finetuning the hyperparameter ρ on ECL leads to changes in E_{MS} loss

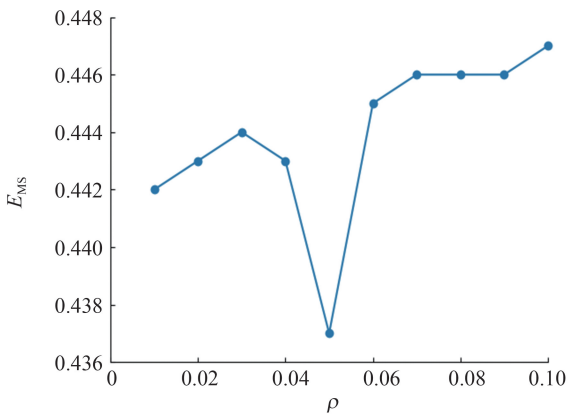

图 2 在 Traffic 数据集上微调超参数 ρ 的 E_{MS} 损失变化
Fig.2 Finetuning the hyperparameter ρ on Traffic leads to changes in E_{MS} loss

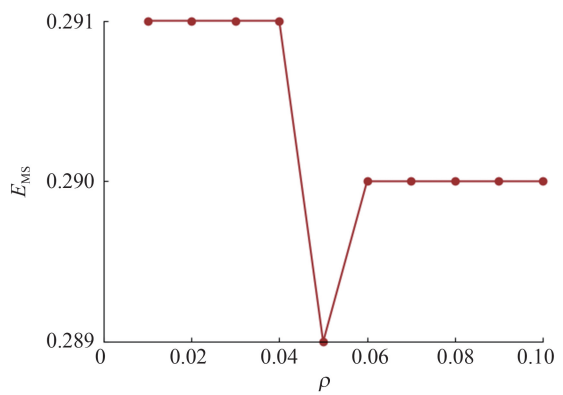

图 3 在 $ETTh_2$ 数据集上微调超参数 ρ 的 E_{MS} 损失变化
Fig.3 Finetuning the hyperparameter ρ on $ETTh_2$ leads to changes in E_{MS} loss

4 结论

本研究关注时间序列的长预测问题,在能源消耗规划和极端天气预报等多种情景下都具有重要意义。为更好地提高预测性能并捕获时间序列的整体趋势,本研究将 Transformer 与季节性趋势分解法相结合,利用季节性趋势分解法捕获时间序列的全局概况。考虑数据隐私问题,本研究利用联邦学习从多个客户端获得一个整体最优预测模型。本研究采用一个基于本地锐度感知的最小化优化器提高全局模型的泛化。与先进的方法相比,本研究方法在 4 个基准数据集上展示了多变量和单变量时间序列预测任务的改进。

参考文献:

[1] 于海东, 刘文彬, 文祥宇. 基于强化学习的电动出租车充电负荷预测[J]. 山东电力技术, 2022, 49(4): 7-14.
YU Haidong, LIU Wenbin, WEN Xiangyu. Electrictaxi charging load forecasting based on reinforcement learning [J]. Shandong Electric Power Technology, 2022, 49 (4): 7-14.

[2] 刘萌, 田雨扬, 谢鑫, 等. 基于模型预测控制的空气源热泵负荷目标温度控制策略[J]. 山东电力技术, 2022, 49(12): 53-59.
LIU Meng, TIAN Yuyang, XIE Xin, et al. Load target temperature control strategy of air source heat pump based on model predictive control[J]. Shandong Electric Power Technology, 2022, 49(12): 53-59.

[3] 路宽, 曲建璋, 高嵩, 等. 基于变分推断的超短期风电功率预测[J]. 山东电力技术, 2023, 50(4): 13-21.
LU Kuan, QU Jianzhang, GAO Song, et al. Ultra-short-term wind power prediction based on variational inference [J]. Shandong Electric Power Technology, 2023, 50 (4): 13-21.

[4] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[J]. Advances in Neural Information Processing Systems, 2017, 30: 5998-6008.

[5] ZHOU H, ZHANG S, PENG J, et al. Informer:beyond efficient transformer for long sequence time-series forecasting[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Palo Alto, USA: AAAI, 2021: 11106-11115.

[6] WU H, XU J, WANG J, et al. Autoformer:decomposition Transformers with auto-correlation for long-term series forecasting[J]. Advances in Neural Information Processing Systems, 2021, 34: 22419-22430.

[7] CLEVELAND R B, CLEVELAND W S, MCRAE J E, et al. STL: a seasonal-trend decomposition[J]. Journal of Official Statistics, 1990, 6(1): 3-73.

[8] WEN Q, GAO J, SONG X, et al. RobustSTL:a robust seasonal-trend decomposition algorithm for long time series[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Honolulu, USA: AAAI, 2019: 5409-5416.

[9] ORESHKIN B N, CARPOV D, CHAPADOS N, et al. N-BEATS: neural basis expansion analysis for interpretable time series forecasting [C]//International

Conference on Learning Representations. New Orleans, USA: ICLR, 2019: 1-31.

[10] MCMAHAN B, MOORE E, RAMAGE D, et al. Communication-efficient learning of deep networks from decentralized data[C]// Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS). Fort Lauderdale, Flordia, USA: JMLR, 2017: 1273-1282.

[11] FORET P, KLEINER A, MOBAHI H, et al. Sharpness-awareminimization for efficiently improving generalization [C]//International Conference on Learning Representations. Addis Ababa, Ethiopia: ICLR, 2020: 1-19.

[12] ZHOU T, MA Z, WEN Q, et al. FEDformer:frequency enhanced decomposed Transformer for long-term series forecasting[C]//International Conference on Machine Learning. Baltimore, USA: PMLR, 2022: 27268-27286.

[13] RANGAPURAM S S, SEEGER M W, GASTHAUS J, et al. Deep state space models for time series forecasting [J]. Advances in Neural Information Processing Systems, 2018, 31: 7785-7794.

[14] SALINAS D, FLUNKERT V, GASTHAUS J, et al. DeepAR:probabilistic forecasting with autoregressive recurrent networks[J]. International Journal of Forecasting, 2020, 36(3): 1181-1191.

[15] SUTSKEVER I, VINYALS O, LE Q V. Sequence to sequence learning with neural networks[J]. Advances in Neural Information Processing Systems, 2014, 27: 3104-3112.

[16] CHO K, VAN MERRIËNBOER B, GULCEHRE C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[EB/OL]. (2014-09-03) [2023-11-07]. https://arxiv.org/abs/1406.1078.

[17] ARIYO A A, AEDWUMI A O, AYO C K. Stock price prediction using the ARIMA model[C]//Proceedings of the 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation. Cambridge, UK: IEEE, 2014: 106-112.

[18] TAYLOR S J, LETHAM B. Forecasting at scale[J]. The American Statistician, 2018, 72(1): 37-45.

[19] DEVLIN J, CHANG M W, LEE K, et al. BERT:pre-training of deep bidirectional Transformers for language understanding[EB/OL]. (2019-05-24) [2023-11-07]. https://arxiv.org/abs/1810.04805.

[20] HUANG C Z A, VASWANI A, USZKOREIT J, et al. Music Transformer: generating music with long-term structure [C]//International Conference on Learning Representations. New Orleans, USA: ICLR, 2019: 1-15.

[21] DOSOVITSKIY A, BEYER L, KOLESNIKOV A, et al. Animage is worth 16×16 words: Transformers for image recognition at scale[C]//International Conference on Learning Representations. [S.l.]: ICLR, 2021: 1-21.

[22] RAO Y, ZHAO W, ZHU Z, et al. Global filter networks for image classification[J]. Advances in Neural Information Processing Systems, 2021, 34: 980-993.

[23] LI S, JIN X, XUAN Y, et al. Enhancing the locality and breaking the memory bottleneck of Transformer on time series forecasting[J]. Advances in Neural Information Processing Systems, 2019, 32: 5243-5253.

[24] WANG S, LI B Z, KHABSA M, et al. Linformer:self-attention with linear complexity [EB/OL]. (2020-06-14) [2023-11-07]. https://arxiv.org/abs/2006.04768.

[25] KAIROUZ P, MCMAHAN H B, AVENT B, et al. Advances and open problems in federated learning [J]. Foundations and Trends in Machine Learning, 2021, 14(1/2): 1-210.

[26] MOHRI M, SIVEK G, SURESH A T. Agnostic federated learning[C]//International Conference on Machine Learning. California, USA: PMLR, 2019: 4615-4625.

[27] LI T, SAHU A K, ZAHEER M, et al. Federated optimization in heterogeneous networks[C]//Proceedings of Machine Learning and Systems. Austin, USA: MLSys, 2020: 429-450.

[28] VENKATESWARAN P, ISAHAGIAN V, MUTHUSAMY V, et al. FedGen: generalizable federated learning for sequential data[C]//Proceedings of the 2023 IEEE 16th International Conference on Cloud Computing (CLOUD). Chicago, USA: IEEE, 2023: 308-318.

[29] ZONG L, XIE Q, ZHOU J, et al. FedCMR:federated cross-modal retrieval[C]//Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, USA: ACM, 2021: 1672-1676.

[30] LI Q, HE B, SONG D. Model-contrastive federated learning[C]//Proceedings of the 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition. Nashville, USA: IEEE, 2021: 10713-10722.

[31] DURMUS A E, YUE Z, RAMON M, et al.Federated learning based on dynamic regularization[C]//International Conference on Learning Representations. [S.l.]: ICLR, 2021: 1-36.

[32] YAO D, PAN W, DAI Y, et al. Local-global knowledge distillation in heterogeneous federated learning with non-IID data[EB/OL]. (2021-09-13) [2023-11-07]. https://arxiv.org/abs/2107.00051.

[33] WANG J, TANTIA V, BALLAS N, et al. SlowMo:improving communication-efficient distributed SGD with slow momentum[C]//International Conference on Learning Representations. Addis Ababa, Ethiopia: ICLR, 2020: 1-27.

[34] REDDI S J, CHARLES Z, ZAHEER M, et al. Adaptivefederated optimization [C]//International Conference on Learning Representations. [S.l.]: ICLR, 2021: 1-38.

[35] KARIMIREDDY S P, JAGGI M, KALE S, et al. Breaking the centralized barrier for cross-device federated learning[J]. Advances in Neural Information Processing Systems, 2021, 34: 28663-28676.

[36] KHANDURI P, SHARMA P, YANG H, et al. STEM:a stochastic two-sided momentum algorithm achieving near-optimal sample and communication complexities for federated learning[J]. Advances in Neural Information Processing Systems, 2021, 34: 6050-6061.

[37] LAKSHMINARAYANAN B, PRITZEL A, BLUNDELL C. Simple and scalable predictive uncertainty estimation using deep ensembles[J]. Advances in Neural Information Processing Systems, 2017, 30: 6402-6413.

[38] WOODWORTH B, GUNASEKAR S, LEE J D, et al. Kernel and rich regimes in overparametrized models [C]//Conference on Learning Theory. Texas, USA: PMLR, 2020: 3635-3673.

[39] QU Z, LI X, DUAN R, et al. Generalized federated learning via sharpness aware minimization[C]//International Conference on Machine Learning. Baltimore, USA: PMLR, 2022: 18250-18280.

[40] CHAUDHARI P, CHOROMANSKA A, SOATTO S, et al. Entropy-SGD:biasing gradient descent into wide valleys[J]. Journal of Statistical Mechanics: Theory and Experiment, 2019, 2019(12): 124018.

[41] KARIMIREDDY S P, KALE S, MOHRI M, et al. SCAFFOLD: stochastic controlled averaging for federated learning[C]//International Conference on Machine Learning. Texas, USA: PMLR, 2020: 5132-5143.

[42] LI T, SANJABI M, BEIRAMI A, et al. Fairresource allocation in federated learning[C]//International Conference on Learning Representations. Addis Ababa, Ethiopia: ICLR, 2020: 1-27.

[43] REISIZADEH A, FARNIA F, PEDARSANI R, et al. Robust federated learning:the case of affine distribution shifts[J]. Advances in Neural Information Processing Systems, 2020, 33: 21554-21565.

[44] KINGMA D P, BA J. Adam: a method for stochastic optimization[EB/OL]. (2017-01-30)[2023-11-07]. https://arxiv.org/abs/1412.6980.

[45] KITAEV N, KAISER L, LEVSKAYA A. Reformer:the efficient transformer[C]//International Conference on Learning Representations. Addis Ababa, Ethiopia: ICLR, 2020: 1-12.

（编辑:孙亚彤）

（上接第100页）

[19] WANG S, DOU Z, ZHU Y. Heterogeneous graph-based context-aware document ranking[C]//Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining. Singapore: ACM, 2023: 724-732.

[20] SUN L, YE J, PENG H, et al. A self-supervised Riemannian GNN with time varying curvature for temporal graph learning[C]//Proceedings of the 31st ACM International Conference on Information & Knowledge Management. Atlanta, USA: ACM, 2022: 1827-1836.

[21] MACAVANEY S, TONELLOTTO N, MACDONALD C. Adaptive re-ranking with a corpus graph[C]//Proceedings of the 31st ACM International Conference on Information & Knowledge Management. Atlanta, USA: ACM, 2022: 1491-1500.

[22] ZHOU X, WANG J, LIU Y, et al. Inductive graph transformer for delivery time estimation[C]// Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining. Singapore: ACM, 2023: 679-687.

[23] TONG P, ZHANG Q, YAO J. Leveraging domain context for question answering over knowledge graph[J]. Data Science and Engineering, 2019, 4: 323-335.

[24] PRESS W H, TEUKOLSKY S A, VETTERLING W T, et al. Numerical recipes: the art of scientific computing [M]. New York, USA: Cambridge University Press, 2007.

[25] R·柯朗, D·希尔伯特. 数学物理方法[M]. 钱敏, 郭敦仁, 译. 北京: 科学出版社, 2011: 57-58.

[26] GAUTSCHI W. Orthogonal polynomials: computation and approximation[M]. Cambridge, UK: OUP Oxford, 2004.

[27] CHENEY E W, LIGHT W A. A course in approximation theory[M]. Providence, USA: AMS, 2009.

[28] LIM D, HOHNE F, LI X, et al. Large scale learning on non-homophilous graphs: new benchmarks and strong simple methods[C]//Advances in Neural Information Processing Systems. [S.l.]: MIT, 2021: 20887-20902.

（编辑:孙亚彤）