计 算 机 研 究 与 发 展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202440476
61(10): 2649−2661, 2024

基于特征约束和自适应损失平衡的机器遗忘方法

殷昱煜[1,2] 吴广强[1] 李尤慧子[1,2] 王鑫雨[1] 高洪皓[3,4]
[1](杭州电子科技大学计算机学院 杭州 310018)
[2](复杂系统建模与仿真教育部重点实验室(杭州电子科技大学)杭州 310018)
[3](上海大学计算机工程与科学学院 上海 200444)
[4](嘉泉大学计算机工程系 韩国城南市 461701)
(yinyuyu@hdu.edu.cn)

A Machine Unlearning Method via Feature Constraint and Adaptive Loss Balance

Yin Yuyu[1,2], Wu Guangqiang[1], Li Youhuizi[1,2], Wang Xinyu[1], and Gao Honghao[3,4]
[1](*School of Computer Science*, *Hangzhou Dianzi University*, *Hangzhou* 310018)
[2](*Key Laboratory of Complex Systems Modeling and Simulation* (*Hangzhou Dianzi University*), *Ministry of Education*, *Hangzhou* 310018)
[3](*School of Computer Engineering and Science*, *Shanghai University*, *Shanghai* 200444)
[4](*Department of Computer Engineering*, *Gachon University*, *Seongnam*, *Republic of Korea* 461701)

Abstract With the accelerated advancement of digitization, data elements have become the core driving force for the operation of modern society. However, at the same time, data security issues have become increasingly prominent, with frequent occurrences of data breaches and privacy violations, causing serious losses to individuals, organizations, and even countries. Against this backdrop, the security of data elements has become the focus of attention from all sectors of society, and the issue of data privacy protection in deep learning models has also attracted widespread attention. Among them, machine unlearning, as a key technology for protecting user's privacy, aims to enable models to remove the influence of specific data while maintaining generalization performance for remaining data, providing an effective solution for protecting the security of data elements in deep learning models. Existing machine unlearning methods are mainly divided into two categories: exact unlearning and approximate unlearning. However, exact unlearning methods need to intervene in the original training process of the models, while approximate unlearning methods find it difficult to strike a balance between unlearning performance and model generalization ability. To address these issues, we propose an approximate unlearning framework based on feature constraints and adaptive loss balancing. We adopt a "forgetting-recovering" machine unlearning framework. First, for the "forgetting" process, in order to mimic the feature outputs of retrained models for the forgetting samples, we use a randomly initialized model that has not been trained on the forgetting samples to guide the feature outputs of the unlearning model, constraining forgetting at the feature level to avoid easily obtaining forgetting data information from the model. Then, a small amount of data is used for fine-tuning to "recover" the generalization performance of models on the remaining data. At the same time, we regard the above machine unlearning framework as a multi-task optimization problem and introduce adaptive loss balance to automatically balance the "forgetting" and "recovering" tasks, preventing the model from "over-forgetting" or "over-recovering", so that the "forgetting" and "recovering" tasks can be trained

收稿日期: 2024-06-03; **修回日期**: 2024-07-19
基金项目: 国家自然科学基金项目(62272140); 浙江省自然科学基金项目(LY22F020018); 浙江省"尖兵""领雁"研发公关计划(2024C01166)
This work was supported by the National Natural Science Foundation of China (62272140) , the Natural Science Foundation of Zhejiang Province (LY22F020018), and the "Pioneer" and "Leading Goose" Research and Development Program of Zhejiang Province (2024C01166).
通信作者: 李尤慧子(huizi@hdu.edu.cn)

relatively balanced and steadily. Extensive experiments on 3 image classification datasets show that our method can effectively forget the forgetting data and achieve optimal performance in multiple metrics.

Key words data element security; machine unlearning; feature constraints; multi-task optimization; adaptive loss balance

摘 要 随着数字化进程的加速推进,数据要素已成为现代社会运转的核心驱动力.由于深度学习模型训练需要大量数据作为输入,其数据隐私保护问题也愈发重要.机器遗忘技术使模型能够删除特定数据的影响,同时保持对剩余数据的泛化性能,为深度学习模型中的数据要素安全保护提供了有效的解决方案.现有的机器遗忘方法主要分为精确遗忘和近似遗忘2类,但前者需要干预模型原始训练流程,后者则在遗忘效果和模型泛化能力之间难以找到平衡点.为此,提出了一种基于特征约束和自适应损失平衡的近似遗忘框架.首先,对于“遗忘”过程,使用同样未经过遗忘样本训练的随机模型作为教师来引导遗忘模型的特征输出,实现模型对数据要素在特征层面的遗忘.然后,采用少量剩余数据进行微调训练,来“恢复”模型对于其他数据的泛化性能.将上述机器遗忘框架视为一个多任务优化问题,在“遗忘”和“恢复”2个任务中引入自适应损失平衡,实现任务的稳步训练.以卷积神经网络模型为例,在3个公开数据集上对比了UNSIR等多种基线方法,实验结果表明,该方法构建的遗忘模型不仅保障了机器遗忘效果,在剩余数据的准确率、时间开销、预测结果分布等指标上优于同类方法,更加有效地保护了模型的泛化性能.

关键词 数据要素安全;机器遗忘;特征约束;多任务优化;自适应损失平衡

中图法分类号 TP391

在当今数字化和信息化的世界中,数据不仅仅是信息时代的核心资源,更是国家安全、个人隐私和商业机密的重要载体.数据要素安全已经成为国家、组织和个人关注的焦点,也是信息技术发展中一个非常重要且亟待解决的问题.一方面,随着大数据和人工智能技术的应用,越来越多的数据被收集和存储,这带来了个人隐私暴露和数据泄漏的风险[1].另一方面,依法人权组织也在要求企业提供数据删除和遗忘的权利,以帮助数据主体控制和管理自己的数据,避免信息的非法及非授权使用[2].但是,传统的数据库技术很难实现真正意义上的数据删除,因为待删除数据可能已经被用于训练人工智能模型,而模型会“记住”这些数据,模型中仍然存在隐藏的数据影响,这使得数据彻底删除变得更加困难.因此,如何更好地满足数据主体隐私需求,精准保障数据要素安全,是当前数据隐私保护研究的重中之重.

机器遗忘(machine unlearning)技术作为一种新兴的数据处理方法,为数据要素安全提供了有力支撑.随着深度学习、人工智能等技术的发展和应用,例如人脸识别[3]、图像生成[4]、自然语言处理[5]等方面,模型的规模不断扩大,所需的训练数据规模也随之增加.模型中包含训练集的细节,会对用户隐私造成一定威胁.研究表明,我们可以从一个已训练好的深度学习模型中恢复部分训练数据(如人脸数据)[6].机器遗忘技术旨在满足用户数据隐私保护方面的个性化需求,即当用户提出“删除数据”的请求时,模型能够迅速且有效地移除特定数据的影响,并维持其在剩余数据集上的泛化性能.这意味着即使模型曾经使用过这些数据,但在用户提出删除请求后,这些数据将不再对模型的预测或决策产生任何影响.这极大地降低了数据泄露和滥用的风险,为用户和企业提供了更加安全灵活的数据要素流通环境.此外,随着用户隐私保护意识的不断增强和相关法律法规的逐步完善,机器遗忘技术在数据要素安全保护方面的应用也越来越多.例如,在欧盟提出的《通用数据保护条例》(General Data Protection Regulation, GDPR)[7]中,明确规定了用户的“被遗忘权”,即用户有权要求数据控制者删除关于其个人数据的记录.机器遗忘技术为这一权利的彻底实现提供了技术支持.

最简单直接的机器遗忘方法就是用剩余数据集重新训练一个模型,但这会导致极大的资源开销,特别是在遗忘请求频繁的情况下.例如,现有的推荐系统常利用用户偏好进行个性化推荐,但随着时间的变化或者潮流的兴起,用户的兴趣和偏好可能会改变,因此,需要删除旧数据的影响以提升推荐准确率.图1描述了机器遗忘的基本流程.当遗忘请求来临时,机器遗忘算法将已经训练好的原始模型作为输入,利用遗忘数据集和剩余数据集生成更新后的遗忘模型.一个理想的机器遗忘方法产生的模型应该与用剩余数据重新训练的模型没有区别.

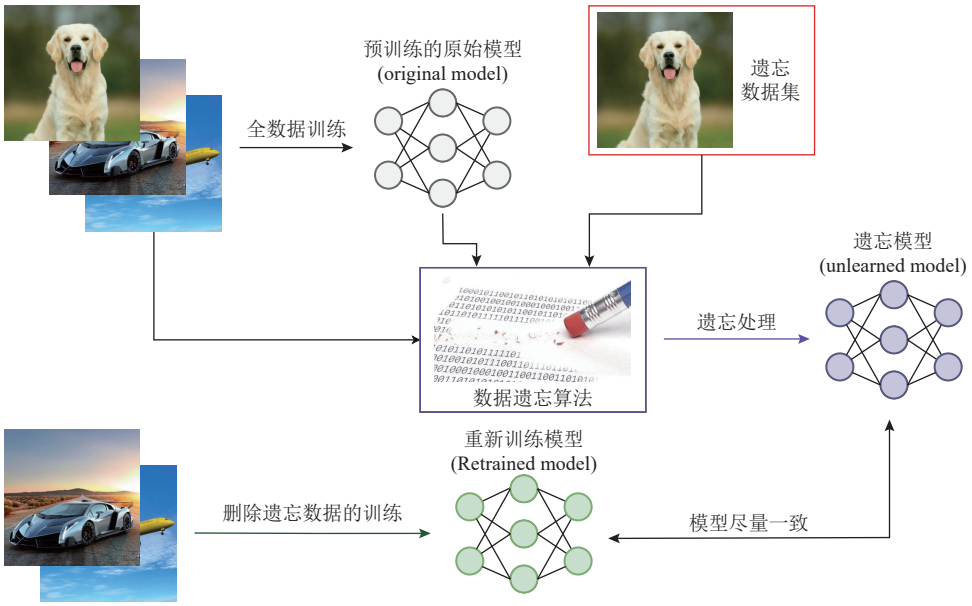


Fig. 1 Illustration of machine unlearning technology

图 1 机器遗忘技术示意图

现有的机器遗忘方法可以分为精确遗忘[8-10]和近似遗忘[11-16]. 精确遗忘方法通常需要巨大的时间开销, 且会干预模型原始的训练流程[16], 最近的研究主要集中于近似遗忘方法. 基于噪声的方法[11-13]虽然实现了近似遗忘, 但添加的噪声会在很大程度上损害模型在剩余数据上的泛化性能. 为了保障模型性能, 文献 [14–15] 采用“遗忘—恢复”的两阶段框架来实现机器遗忘, 通过遗忘后的微调或知识蒸馏来恢复模型. 但这类方法会造成模型对遗忘数据产生异常的预测准确率, 例如, 基于文献 [15] 提出的方法进行遗忘后的模型对遗忘数据的预测准确率为 10%, 而重新训练的模型(基准模型)对该数据的预测准确率仅为 0%. 我们注意到, 这些机器遗忘方法只是在模型的分类输出(logits)层面实现对样本数据的遗忘, 而没有考虑模型的特征层面的变化. 这可能会导致“史翠珊效应”[11], 即我们试图隐藏的信息反而更加突出. 例如, 将所有遗忘样本都故意预测为某一个类, 这可能会暴露更多遗忘数据的信息, 容易被侧信道攻击所利用. 综上所述, 现有的方法很难兼顾遗忘效果和模型泛化性能, 尤其是在模型分类准确率方面.

基于上述启发, 我们提出了基于特征约束和自适应损失平衡的“遗忘—恢复”两阶段机器遗忘框架, 使遗忘模型更加关注遗忘数据的特征输出(而不是分类输出), 同时保留模型的泛化性能. 在“遗忘”阶段, 我们将随机初始化的模型作为教师模型来引导遗忘模型的特征输出, 将机器遗忘约束在特征层面, 避免轻易获得或恢复遗忘数据信息. 在“恢复”阶段, 我们利用剩余数据集的 10%~20% 对模型微调, 来恢复模型的泛化能力, 同时避免产生过多的时间开销. 基于“遗忘—恢复”的两阶段机器遗忘方法与多任务学习[17]类似, 需要平衡多个任务的学习, 避免模型在其中一个任务上学习的速率过快, 而没有从另一个任务中学到知识. 具体来说, 我们需要平衡“遗忘”和“恢复”2 个任务, 避免“过度遗忘”或者“过度恢复”. 因此, 我们引入了自适应损失[18], 在训练过程中自动平衡“遗忘”和“恢复”任务, 使模型在完成特征遗忘的同时, 最大化地保留模型的泛化性能. 另外, 我们引入了基于参数重要性的正则化项[19], 将模型中对遗忘数据重要的参数进行较多的更新, 同时保留其他参数, 避免模型在剩余数据上的效果衰退.

本文工作的主要贡献包括 3 个方面:

1)提出了一种基于特征约束和自适应损失平衡的机器遗忘方法. 在“遗忘”阶段, 通过模拟随机初始化模型对遗忘数据的特征输出, 从特征层面约束遗忘模型, 使其包含更少的数据信息. 在“恢复”阶段, 利用剩余数据的 10%~20% 进行微调, 来恢复模型泛化性能.

2)将基于“遗忘—恢复”的两阶段机器遗忘框架抽象为一个多任务学习问题, 为了更好地平衡“遗忘”任务和“恢复”任务, 引入了自适应损失平衡, 在训练过程中动态地调整遗忘损失和恢复损失, 防止“过度遗忘”或“过度恢复”.

3)与 UNSIR, Boundary Shrink 等 6 种机器遗忘方法在 CIFAR10, CIFAR100, MNIST 数据集上进行实验分析验证，结果表明我们的方法可以在保障机器遗忘效果的同时更好地保护模型的泛化性能，消融实验进一步证明了各模块的有效性.

1 相关工作

现有的机器遗忘方法主要包括：精确遗忘和近似遗忘.

精确遗忘能够确保将遗忘数据对模型的影响完全消除，它可以通过加速模型重新训练的方式实现. Bourtoule 等人[8]提出了一种通用的基于集成学习的机器遗忘框架 SISA，将原始模型划分为多个不相交的子模型，模型的输出由多个子模型聚合而得，当机器遗忘请求到达时，只需要重新训练受影响的子模型. 类似地，Yan 等人[9]设计了 ARCANE，与 SISA 不同的是，该方法按类别划分子模型，利用 One-class 分类技术提高子模型的准确性. 此外，ARCANE 还对每个子数据集进行了预处理，以进一步加速模型的再训练. 与上述将模型分为隔离的子模型不同，Dukler 等人[10]提出的 SAFE 将模型划分为相互影响的子模型. SAFE 引入了“碎片图”的概念，它允许每个分片在训练期间整合来自其他分片的有限信息. 当遗忘请求到来时，SAFE 会将遗忘数据点直接影响和间接影响的分片都重新训练，在遗忘成本的增加和精度的提高之间进行了权衡. 精确遗忘方法不包含遗忘数据的任何信息，因此也不包含遗忘数据的特有特征，实现了特征级别的遗忘，但它们干预了模型原始的训练流程，这可能会损伤模型，进而不能适用于原始的应用需求，例如 SISA 中，划分出的独立子模型的泛化性能可能与原始模型相差较大[16]. 另外，机器遗忘算法更常用的情况是将已经训练好的模型作为起点，并消除所请求数据的影响[11].

近似遗忘通过更新模型参数消除遗忘数据对模型的影响，使遗忘后模型与重新训练的模型没有显著差异. 部分研究通过添加噪声来实现近似遗忘. Golatkar 等人[11]提出了 Fisher Forgetting，利用 Fisher 信息矩阵计算来破坏信息的最佳噪声，从而达到遗忘数据的效果. Huang 等人[12]提出了一种误差最小化噪声，它会欺骗模型，使模型认为从这些样例中学习不到任何知识. 与“不可学习的样本”相反，Tarun 等人[13]提出的 UNSIR 使用很高的学习率为遗忘数据学习错误最大化噪声矩阵，从而诱导模型急剧遗忘. 基于噪声的方法可以有效地使模型完成遗忘，但会在很大程度上降低模型在剩余数据上的泛化性能. 因此，近期研究[14-15]采用“遗忘—恢复”的两阶段框架，通过遗忘后的微调或知识蒸馏[20]来恢复模型性能. 这种方法虽然可以显著提高模型在剩余数据上的泛化性能，但会使模型对遗忘数据产生异常的预测准确率，例如，基于文献 [15] 提出的方法进行遗忘后的模型对遗忘数据的预测准确率为 10%，而重新训练的模型(基准模型)对该数据的预测准确率仅为 0%. 综上所述，现有的近似遗忘方法很难兼顾遗忘效果和模型泛化性能，尤其是在模型分类准确率方面.

本文提出的基于特征约束和自适应损失平衡的机器遗忘方法(machine unlearning method via feature constraint and adaptive loss balance, FCALB)，通过特征遗忘避免轻易获得或恢复遗忘数据信息. 同时，我们将“遗忘—恢复”两阶段机器遗忘流程构建为一个多任务学习问题，通过加入自适应损失平衡，动态地调整“遗忘”和“恢复”任务的训练速率，防止“过度遗忘”或“过度恢复”现象. 该方法在实现对目标数据的遗忘的同时保留了模型的泛化性能，且在不干预模型的原始训练流程的情况下以较快的速度完成了机器遗忘.

2 本文方法

本节主要介绍基于特征约束和自适应损失平衡的机器遗忘方法，如图 2 所示，该方法包括“遗忘”和“恢复”2 个任务. 在“遗忘”任务中，本文方法更加关注模型的特征层面，使遗忘模型学习随机初始化模型的特征输出，防止轻易地重建遗忘数据的信息. 为了在遗忘的同时保留模型在剩余数据上的性能，在“恢复”任务中利用每个类别的剩余数据的 10%~20% 进行微调来恢复模型的泛化性能. 同时，本文方法引入了自适应损失权重来平衡“遗忘”和“恢复”2 个任务学习的趋势. 最后，我们利用 Fisher 信息矩阵进行了参数正则化，使模型中对遗忘数据样本分类重要的参数尽可能多地更新，从而更好地约束“遗忘”和“恢复”任务.

2.1 问题定义

本文研究的是深度神经网络中的机器遗忘问题. 设训练数据集为 $D=\{(x_i,y_i)\}_{i=1}^{n}$，其中 x_i 表示训练样本，y_i 表示对应的标签，标签值落在范围 $\{0,1,\cdots,K-1\}$ 内，K 为标签类的总数. 我们将待遗忘数据定义为 $D_{\mathrm{f}}\subseteq D$，希望保留的剩余数据为 $D_{\mathrm{r}}=D\backslash D_{\mathrm{f}}$. 机器遗忘的目标

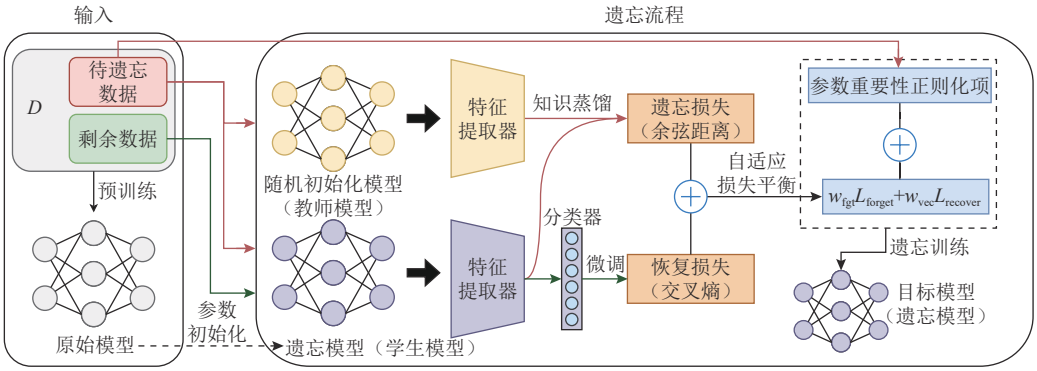


Fig. 2 Framework of our proposed method

图 2 本文方法框架

是设计一个算法，使得遗忘后模型 M_{unlearn} 不包含 D_{f} 的任何信息，并且保留模型在 D_{r} 上的性能. 理想情况下，M_{unlearn} 应该与仅在 D_{r} 上训练的模型 M_{retrain} 是相似的，即

$$M_{\text{unlearn}}(D_{\text{r}}) \approx M_{\text{retrain}}(D_{\text{r}}). \tag{1}$$

需要注意的是，机器遗忘并不是使模型对 D_{f} 进行简单地误分类，对于 M_{retrain} 而言，大多数遗忘样本被预测为特定的类，而不是随机的错误类别[16]. 因此，我们应该更加关注模型的特征层面，避免只是在分类层面的机器遗忘.

2.2 基于特征约束的“遗忘”过程

通常，在深度神经网络（DNN）中，网络的较低层可以视为特征提取器，而顶层可以视为线性分类器. 如卷积神经网络（CNN）中，对于分类问题，最后一个卷积层的特征图通过量化后与全连接层连接，再接一个 softmax 逻辑回归分类层得到预测结果. 设网络共有 L 层，我们可以将最后的 softmax 全连接层视为线性分类器，前面的由多个卷积层组成的 $L-1$ 层视为特征提取器. 由于顶层分类器的线性性质，从顶层隐藏层中提取的特征通常是线性可分的，于是，我们可以通过比较 2 个不同的 DNN 特征提取器的输出来判断 2 个网络提取的特征是否相似.

在机器遗忘中，我们希望 M_{unlearn} 对于 D_{f} 的特征输出与 M_{retrain} 是相似的，但由于 M_{retrain} 是未知的，我们无法通过 M_{retrain} 作为教师模型引导 M_{unlearn} 的特征输出. 因此，本文提出了一种替代方案：使用随机初始化的模型 M_{random} 作为教师模型，使 M_{unlearn} 对于 D_{f} 的特征输出与 M_{random} 相似. 这是因为，M_{random} 和 M_{retrain} 都不包含 D_{f} 的先验知识，也就是说，它们“从未见过”要遗忘的数据，M_{random} 和 M_{retrain} 都没有经过 D_{f} 进行训练，因此它们都不包含 D_{f} 的特征. 虽然也可以利用随机初始化模型对输出的概率分布进行蒸馏[15,21]，但这与随机标签方法类似，只是使模型对 D_{f} 误分类，可能反而暴露更多 D_{f} 的信息. M_{retrain} 对 D_{f} 的输出存在一定的不确定性，但并不是随机预测.

具体地，如图 2 所示，本文将遗忘模型 M_{unlearn} 的权重参数 $\theta^{(\text{u})}$ 初始化为原始模型 M_{origin} 的权重参数 $\theta^{(\text{o})}$，确保模型初始阶段在剩余数据集上有良好的性能. 同时，将随机初始化的模型 M_{random} 作为教师模型，将待遗忘数据 D_{f} 作为输入，引导 M_{unlearn} 在 D_{f} 上的特征输出近似于 M_{random} 的特征输出. 本文采用余弦距离作为相似性度量指标：

$$\cos(\boldsymbol{v}_1,\boldsymbol{v}_2)=\frac{\langle \boldsymbol{v}_1,\boldsymbol{v}_2\rangle}{\|\boldsymbol{v}_1\|\cdot\|\boldsymbol{v}_2\|}. \tag{2}$$

2 个向量的余弦值越高，则向量间的夹角越小，相似性越高. 因此，“遗忘”过程的损失函数可以表示为：

$$L_{\text{fgt}}=1-\cos\left(\boldsymbol{f}_{L-1}\left(x_{\text{f}};\theta^{(\text{r})}\right),\boldsymbol{f}_{L-1}\left(x_{\text{f}};\theta^{(\text{u})}\right)\right), \tag{3}$$

其中，x_{f} 表示来自 D_{f} 的训练样本，$\theta^{(\text{u})}$ 表示遗忘模型 M_{unlearn} 的权重，$\theta^{(\text{r})}$ 表示随机初始化模型 M_{random} 的权重，L 为模型的总层数，$\boldsymbol{f}_{L-1}$ 表示模型的第 $L-1$ 层输出的特征向量.

2.3 基于自适应损失平衡的“恢复”过程

“遗忘”过程中特征级别的遗忘使 M_{unlearn} 在遗忘数据上的特征输出与 M_{random} 相似，但由于模型对于不同类别的数据提取的特征存在一定的相似性（共有特征），导致 M_{unlearn} 在剩余数据上的表现有所下降，且下降程度会随着模型架构复杂度的增加而减小.

为了恢复 M_{unlearn} 在 D_{r} 上的性能，我们利用小部分剩余数据 $D_{\text{r_sub}}\subseteq D_{\text{r}}$ 来微调 M_{unlearn}，并用交叉熵作为损失函数：

$$L_{\text{rec}}=-\sum_{i=0}^{n} y_i\log(p(x_i)), \tag{4}$$

其中，x_i 和 y_i 分别为来自 $D_{\text{r_sub}}$ 的样本和标签.

本文方法的损失函数可以由遗忘和恢复 2 个阶

段的损失函数加权得到：

$$L_{\text{total}} == w_{\text{fgt}}L_{\text{fgt}} + w_{\text{rec}}L_{\text{rec}}. \tag{5}$$

通常情况下，$D_{\text{r}} \gg D_{\text{f}}$，数据的不平衡性会导致难以同时优化2个不同的目标[22]，造成其中一个任务训练的速度很快，而模型并没有从另一个任务中学到任何东西. 为了更好地平衡“遗忘”和“恢复”2个目标，本文将机器遗忘看作一个多任务学习[17]的优化问题，引入了自适应损失平衡[18]来动态地更新2个损失的权重，同时优化“遗忘”和“恢复”2个任务.

具体地，遗忘损失的权重可以表示为：

$$w_{\text{fgt}}(t) = \eta_{w'} r_{\text{fgt}}(t) + (1-\eta_{w'}) w_{\text{fgt}}(t-1), \tag{6}$$

其中，$w_{\text{fgt}}(t)$表示在模型训练迭代的第t步时遗忘损失$L_{\text{fgt}}(t)$的权重，$\eta_{w'}$为自适应损失权重更新的学习率，$r_{\text{fgt}}(t)$为相对损失权重，它描述了在时刻t $L_{\text{fgt}}(t)$的相对变化速率，具体表示为：

$$r_{\text{fgt}}(t) = \frac{\tilde{L}_{\text{fgt}}(t)}{\tilde{L}_{\text{fgt}}(t) + \tilde{L}_{\text{rec}}(t)}, \tag{7}$$

$$\tilde{L}_{\text{fgt}}(t) = \frac{L_{\text{fgt}}(t)}{L_{\text{fgt}}(0)}, \tag{8}$$

$$\tilde{L}_{\text{rec}}(t) = \frac{L_{\text{rec}}(t)}{L_{\text{rec}}(0)}, \tag{9}$$

其中，$\tilde{L}_{\text{fgt}}(t)$和$\tilde{L}_{\text{rec}}(t)$分别反映了在训练的第$t$步时遗忘损失$L_{\text{fgt}}(t)$和恢复损失$L_{\text{rec}}(t)$相对于初始时刻$L_{\text{fgt}}(0)$和$L_{\text{rec}}(0)$的变化幅度，它们的值越小，则说明对应任务的训练速度越快.

同样地，恢复损失的权重可以表示为：

$$w_{\text{rec}}(t) = \eta_{w} r_{\text{rec}}(t) + (1-\eta_{w}) w_{\text{rec}}(t-1), \tag{10}$$

$$r_{\text{rec}}(t) = \frac{\tilde{L}_{\text{rec}}(t)}{\tilde{L}_{\text{fgt}}(t) + \tilde{L}_{\text{rec}}(t)}. \tag{11}$$

通过自适应损失平衡，在训练的过程中根据损失的变化情况自适应地更新损失权重，使2个任务的损失以较为平衡的方式稳步下降(初始时刻w_{fgt}和w_{rec}均为0.5，确保在训练开始时，2种损失对总损失的贡献是均等的，从而避免模型在初期过度关注某一个特定的损失).

2.4 基于参数重要性的正则化

基于参数重要性的正则化项通过多次更新模型在D_{f}上正确分类的重要关键参数，从而更好地实现遗忘，反之，也可以减少模型在D_{r}上的效果衰退.

首先，通过计算Fisher信息矩阵的对角线来估计参数重要性，Fisher信息矩阵的对角线的值等同于模型损失在最小值附近的二阶导数(即曲率)，反应了模型的参数对于损失的敏感度[23]. Fisher信息矩阵可以从一阶导数进行计算[24]，其第k个对角线元素F_k表示参数θ_k的重要性：

$$F_{ij}(\theta) = E\left[\frac{\partial \ln p(X;\theta)}{\partial \theta_i} \cdot \frac{\partial \ln p(X;\theta)}{\partial \theta_j}\right] = -E\frac{\partial^2 \ln p(X;\theta)}{\partial \theta_i \theta_j}. \tag{12}$$

我们希望模型中对D_{f}分类越重要的参数更新得越多，因此本文将每一层的参数重要性值$\hat{\boldsymbol{\Omega}}_l = \{\boldsymbol{\Omega}_1, \boldsymbol{\Omega}_2, \cdots, \boldsymbol{\Omega}_n\}$线性归一化到[0,1]区间内($\boldsymbol{\Omega}_i = \dfrac{\boldsymbol{\Omega}_i - \text{Min}(\hat{\boldsymbol{\Omega}}_l)}{\text{Max}(\hat{\boldsymbol{\Omega}}_l) - \text{Min}(\hat{\boldsymbol{\Omega}}_l)}$)，然后取反，得到用于训练遗忘模型的参数重要性值$\bar{\boldsymbol{\Omega}}_i = 1 - \boldsymbol{\Omega}_i$，本文利用$\bar{\boldsymbol{\Omega}}_i$约束对$D_{\text{f}}$重要的参数进行较多的更新，正则化项为：

$$R_{\text{imp}}(\theta) = \sum_i \bar{\boldsymbol{\Omega}}_i \left(\theta_i^{\text{u}} - \theta_i^{\text{u}*}\right)^2, \tag{13}$$

其中，i为参数的序号，θ_i^{u}为遗忘模型的第i个参数值，$\theta_i^{\text{u}*}$为初始遗忘模型(也就是原始模型)的第i个参数值.

2.5 目标优化函数及模型训练

本文基于特征蒸馏和自适应损失权重平衡实现了机器遗忘，同时，加入了参数重要性正则化，以更好地擦除D_{f}的“知识”. FCALB的总体损失函数为：

$$L_{\text{total}} = w_{\text{fgt}}L_{\text{fgt}} + w_{\text{rec}}L_{\text{rec}} + \lambda \sum_i R_{\text{imp}}(\theta), \tag{14}$$

其中，w_{fgt}和w_{rec}分别为遗忘模型和恢复模型训练过程中得到的自适应损失权重，λ为限制正则化强度的超参数.

在训练时，通过将可用的剩余数据批次(batch)的数量配置为遗忘批次数量的倍数(*batch-ratio*≥1)来限制$D_{\text{r_sub}}$的数量[25]. 机器遗忘训练的每个批次对应于自适应损失平衡中的每一步t，也就是说，在当前批次的训练中，遗忘损失和恢复损失的权重($w_{\text{fgt}}(t)$，$w_{\text{rec}}(t)$)是基于上一个批次的损失权重($w_{\text{fgt}}(t-1)$，$w_{\text{rec}}(t-1)$)进行计算更新的.

具体的算法流程如算法1所示.

算法1. 基于特征约束和自适应损失平衡的机器遗忘算法.

输入：训练轮次*epochs*，遗忘数据集D_{f}，剩余数据集D_{r}，随机初始化模型M_{random}，遗忘模型M_{unlearn}，当前遗忘模型参数$\theta^{(\text{u})}$，初始遗忘模型参数$\theta^{(\text{u})*}$，原始模型参数$\theta^{(\text{o})}$，迭代步骤t=0，剩余样本比例*batch-ratio*；

输出：遗忘后模型M_{unlearn}.

① 初始化$\theta^{(\text{u})*} = \theta^{(\text{o})}$；

② 计算参数重要性矩阵$\bar{\boldsymbol{\Omega}}_i$；

③ for e in *epochs* do

④ for $(x_{\text{f}}, y_{\text{f}})$ in D_{f} do

⑤ for *batch-iter* (x_r, y_r) in D_r do
⑥ if *batch-iter* ⩾ *batch-ratio* then
⑦ break;
⑧ end if
⑨ 计算M_{unlearn}和M_{random}的特征输出 $\boldsymbol{f}_{L-1}\left(x_f;\theta^{(u)}\right)$, $\boldsymbol{f}_{L-1}\left(x_f;\theta^{(r)}\right)$;
⑩ 根据余弦相似度计算遗忘损失L_{fgt};
⑪ 根据交叉熵计算恢复损失L_{rec};
⑫ 更新遗忘损失和恢复损失权重;
⑬ 根据$\bar{\boldsymbol{\Omega}}_i$计算参数重要性正则化项 $R_{\text{imp}}\left(\theta^{(u)}\right)$;
⑭ 计算总体损失L_{total};
⑮ 根据L_{total}反向传播更新M_{unlearn}参数;
⑯ t++;
⑰ end for
⑱ end for
⑲ end for

3 实验结果与分析

本文在3个图像分类数据集上，从模型效用、时间开销、隐私保护效用等方面，对本文方法进行了评估. 文中涉及的所有实验均在NVIDIA RTX A6000 GPU, 48 GB RAM的机器上使用Python3.8和PyTorch1.12.0[26]实现.

3.1 数据集和预训练模型

我们采用图像分类数据集作为评测数据: CIFAR10[27], CIFAR100[27], MNIST[28]. 其中, CIFAR10分别用AllCNN[29]和ResNet18[30]作为原始模型; CIFAR100使用ResNet50[28]作为原始模型; MNIST使用ResNet18作为原始模型.

3.2 测评指标

机器遗忘方法评测指标包括:

1)准确率. 模型在遗忘数据和剩余数据上的准确率应该与重新训练模型相似.

2)时间开销. 衡量遗忘方法的效率. 遗忘方法的时间开销越小越好, 最大不能超过重新训练模型所消耗的时间.

3)成员推理攻击(membership inference attack, MIA)[31]. MIA的主要目的是判断一个数据样本是否被用于模型训练. 在机器遗忘领域, 常用MIA评估遗忘模型的隐私保护效果[16], 判断遗忘模型是否还包含D_f的信息. 理想情况下, 遗忘模型和重新训练模型有着相近的攻击成功率.

4)输出分布. 我们分析了遗忘模型的输出分布, 特别是对遗忘数据的输出分布, 它并不是随机预测. 遗忘模型的输出分布应尽量与重新训练模型的相似.

3.3 对比方法

为了证明我们的遗忘方法的有效性, 本文模型与6个基线模型进行了比较:

1)重新训练(Retrained). 利用D_r从头重新训练, 得到的模型不包含任何D_f的信息. 因此, 重新训练模型为最优遗忘模型, 作为基准参照.

2)微调(Fine-tune)[11]. 在原始模型的基础上, 用较大的学习率和D_r进行训练. 这类似于自然发生的"灾难性遗忘"[19,32], 在没有D_f的情况下进行微调可能会使模型忘记D_f的原始解.

3)随机标签(Random Labels)[11]. 采用"遗忘—恢复"框架, 在"遗忘"阶段, 为D_f的每个样本设置随机错误标签, 在"恢复"阶段, 将它和D_r投入原始模型进行训练.

4)Fisher Forgetting[11]. 利用Fisher信息矩阵计算最优噪声, 将从高斯分布中采样的Fisher噪声添加到模型权重中完成遗忘.

5)UNSIR[13]. 基于D_f计算错误最大化噪声矩阵来完成遗忘, 然后使用D_r的一小部分数据进行微调.

6)Boundary Shrink(BS)[16]. 目前最优的近似遗忘方法. 利用FSGM[33]生成D_f中每个样本的错误类别, 将原始模型的决策边界移动到最近且错误的决策边界中.

3.4 对比试验及结果分析

1)模型效用

对于分类任务而言, 一个有效的机器遗忘算法应该使模型降低对遗忘数据的准确率, 并保留模型在剩余数据上的泛化性能. 如表1所示, 我们首先在CIFAR10数据集的基础上用ResNet18和AllCNN模型对比了不同的机器遗忘方法. 其中, A_r^t和A_f^t表示模型在测试集中的保留数据D_r^t和遗忘数据D_f^t上的准确率. 我们可以看到, 所有遗忘方法都可以将模型在D_f^t上的分类准确率降低, 它们的差异具体体现在对模型泛化性能的保护上. 其中, 基于添加噪声的Fisher Forgetting和UNSIR方法会极大损害模型的效用(ResNet18模型中A_r^t从78.83%降到了71.48%). 采用较大的学习率Fine-tune可以较好地保留模型性能, 但该方法需要较长的时间训练, 尤其是当模型的规模变大时, 例如, 表2中Fine-tune在10个类的遗忘场景中需要训练2 211 s. Random Labels虽然可以将模型在D_f^t上的准确率降为0, 但它其实并没有删除遗忘数

据对模型的影响, 会导致“史翠珊效应”. Boundary Shrink 为目前最好的机器遗忘方法, 但在D_f^t上保留的准确率较高(在 ResNet18 模型上的 5.75% 和 AllCNN 上的 4.90%). 相反, 本文方法降低了模型在D_f^t上的准确率(在 ResNet18 模型上A_f^t仅为 1.10% 和 0%), 同时也在很大程度上保留了模型的泛化性能. 对于 ResNet18 模型, 本文方法的A_r^t与重新训练模型最接近(76.18% 和 79.48%). 此外, 我们发现几乎每个基线模型在 AllCNN 上的结果都比在 ResNet18 好一点, 而本文方法却恰恰相反, 这可能是因为越复杂的模型往往能够学习到更细粒度、类别特有的特征, 而 AllCNN 模型规模较小, 对于不同类别提取的特征重叠的部分较多, 遗忘阶段对模型的损伤较大.

Table 1 Accuracy Comparison on ResNet18 and AllCNN Models

表 1 ResNet18 和 AllCNN 模型上的准确率对比 %

方法	ResNet18				AllCNN			
	1 class		2 classes		1 class		2 classes	
	A_r^t	A_f^t	A_r^t	A_f^t	A_r^t	A_f^t	A_r^t	A_f^t
Origin	77.68	77.06	78.83	72.88	82.08	93.08	81.86	88.37
Retrained	78.36	0	81.31	0	82.42	0	83.19	0
Fine-tune	75.06	0	75.44	0	79.58	0	78.46	1.39
Random Labels	75.95	0	76.15	0	78.11	0	74.88	0
Fisher Forgetting	66.47	0	69.28	0	63.76	0.40	66.29	0.09
UNSIR	72.88	0	71.48	0	78.02	0	79.13	0.15
Boundary Shrink	74.46	2.60	77.03	5.75	76.79	4.90	79.43	1.40
本文方法	76.18	1.10	79.48	0	79.16	0.90	78.50	2.05

Table 2 Accuracy Comparison on ResNet50 Model

表 2 ResNet50 模型上的准确率对比

方法	10			20			40		
	A_r^t/%	A_f^t/%	训练时间/s	A_r^t/%	A_f^t/%	训练时间/s	A_r^t/%	A_f^t/%	训练时间/s
Origin	76.69	78.65	8 185	77.27	75.32	8 185	77.14	76.47	8 185
Retrained	76.94	0	8 572	77.33	0	9 995	76.62	0	10 024
Fine-tune	76.70	0	2 211	78.19	0	2007	80.02	0	1 474
Random Labels	73.12	0	2 439	73.01	0	2 464	71.33	0	2 443
UNSIR	69.35	4.81	93	70.24	1.15	161	75.89	0.05	463
Boundary Shrink	73.92	6.90	471	73.59	7.45	573	73.17	15.93	1 139
本文方法	75.40	3.40	140	76.82	0.20	295	77.67	0	493

因此, 我们还在较大的 ResNet50 模型和 CIFAR-100 数据集上对遗忘方法进行了评估, 如表 2 所示, 本文方法在准确率方面超越了除了 Fine-tune 以外的其他所有方法. 虽然 Fine-tune 对模型泛化性能的保护最优, 但代价是较长的遗忘时间.

2)时间开销

本节从遗忘模型训练时间分析了各个遗忘方法的效率. 如图 3 所示, 本文方法与 Retrained 相比, 分别提升了 24.5 倍(基于 ResNet18 模型和 CIFAR10 数据集)、15.7 倍(基于 AllCNN 模型和 CIFAR10 数据集)、7.8 倍(基于 ResNet18 模型和 MNIST 数据集). 本文方法虽然在遗忘时间方面比 UNSIR 大, 但如表 1 和表 2 所示, UNSIR 会对模型的泛化性能影响较大, 而本文方法可以有效地保护模型的泛化性能. 总的来说, 与这些基线方法相比, 本文方法更加高效和快速. 此外, 由于 Fisher Forgetting 方法的时间开销非常大, 所以没有放在表 2 中做同类比较. Fisher Forgetting 在基于 AllCNN 模型架构的机器遗忘中耗时 3.5×10^3 s, 它的时间开销比重新训练还要长, 不符合机器遗忘方法的需求.

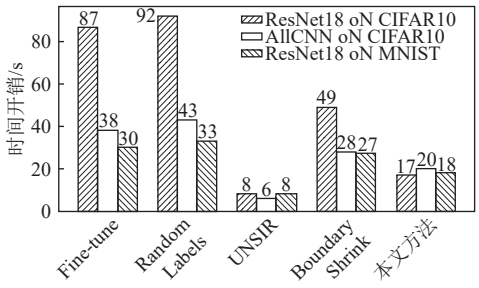


Fig. 3 Comparison of time cost

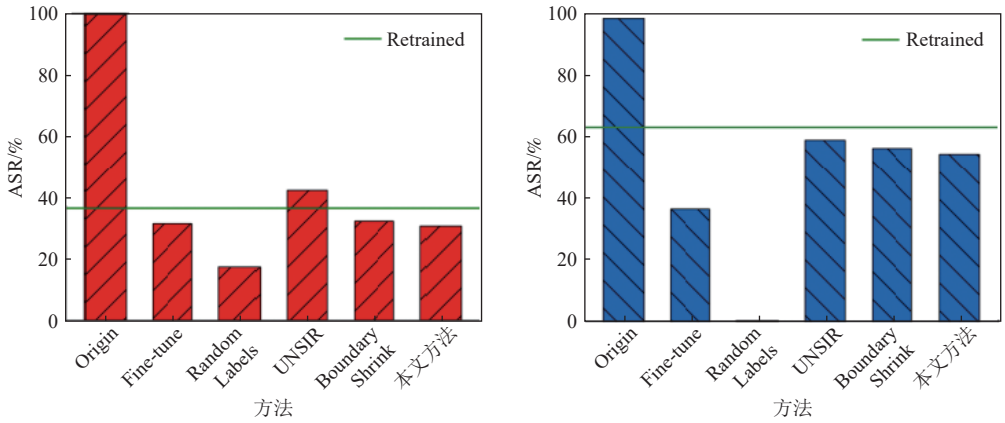
图 3 时间开销对比

从图 3 看出, 本文方法更适用于模型规模较大的情况(所有基线方法在 AllCNN 架构下对 CIFAR10 数据集的遗忘时间都要比 ResNet18 架构下少很多), 因为复杂的模型一般能够提取更加细粒度的特征.

3)隐私保护效果

为了进一步验证机器遗忘方法的隐私保护效果, 本文对遗忘模型进行了成员推理攻击. 成员推理攻击目标是检测一个样本是否被用于模型的训练, 攻击成功率越高, 则说明这个样本很大概率为训练样本. 因此, 我们使用D_f来验证遗忘模型的 MIA 攻击成功率(attack success rate, ASR). ASR 值高, 说明模型几乎没有擦除D_f的信息; ASR 值低, 说明可能发生了“史翠珊效应”, 反而提供给攻击者更多的D_f信息. 理想情况下, 遗忘模型的 ASR 值应该与重新训练模型的 ASR 值接近.

ResNet18 模型在 CIFAR10 和 MNIST 数据集上的实验结果如图 4 所示. 其中, 水平的基准线表示重新



(a) CIFAR10数据集 (b) MNIST数据集

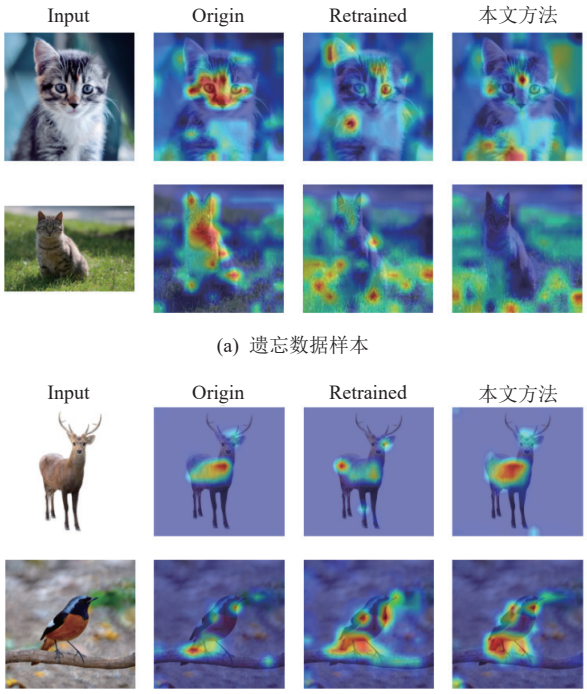
Fig. 4 Comparison of ASR in member inference

图 4 成员推理攻击成功率对比

训练模型的 ASR 值. 我们可以看到, 由于 Origin 没有进行遗忘操作, 所以 ASR 值接近 100%. Fine-tune 在 MNIST 上的 ASR 值与重新训练模型相差较大. Random Labels 在 CIFAR10 和 MNIST 上都表现出很低的 ASR 值, 这说明发生了“史翠珊效应”[17]. 这可能是因为, 它尽可能地使模型对 D_f 分类错误, 而没有真正擦除模型中有关 D_f 的信息. UNSIR、Boundary Shrink 和本文方法的 ASR 值都与 Retrained 的 ASR 值非常接近. UNSIR 通过学习错误最大化噪声, 使模型的遗忘程度较深. Boundary Shrink 模拟了 Retrained 相对于 Origin 的边界移动. 而本文方法学习 Random 对于 D_f 的特征提取, 使模型对其的特征输出接近于从未在 D_f 上训练过的 Retrained, 但 Random 与 Retrained 对于未知数据的特征输出仍然存在一些差距, 这可能是导致本文方法的 ASR 效果稍逊于 UNSIR 和 Boundary Shrink 的主要原因, 如在 MNIST 数据集上, Boundary Shrink 的 ASR 值为 0.566, 本文方法为 0.546. 总的来说, 本文方法仍然可以保证模型的隐私效用, 它的 ASR 值与 Retrained 的 ASR 值非常接近.

4)可视化分析

为了更直观地对比模型在机器遗忘前后的效果, 本文利用 Grad-CAM[34] 以可视化的方式展示模型关注的焦点区域. 图 5 为 Retrained 和本文方法基于 ResNet18 模型和 CIFAR10 数据集的结果. 如图 5(a)所示, 对于遗忘数据(如 CIFAR-10 中的第 3 个类别, 对应的标签为“猫”), Origin 关注物体的面部特征以及形状特征. 本文方法和 Retrained 相似, 不再关注类别的原有特征, 而是将注意力转移到图片的背景中. 对于剩余数据, 如图 5(b)所示, 本文方法保留了原有模型的泛化性能, 与 Retrained 和 Origin 关注的区域相同.



(a) 遗忘数据样本

(b) 剩余数据样本

Fig. 5 Comparison of visual heat maps

图 5 可视化热力图对比

5)预测分布

一个不正确的机器遗忘方法可能会使模型在 D_f 上的准确率降低, 但会存在一些潜在的漏洞(“史翠珊”效应), 更容易暴露 D_f 的信息. 本节通过遗忘模型的输出分布来验证机器遗忘方法的有效性. 图 6 展示了使用不同遗忘方法后模型输出熵的分布密度图(熵代表了不确定性, 模型输出的不确定性越高, 则输出熵值越大). 我们可以发现, 对于 Origin 来说, 由于训练数据为 D_r 和 D_f, 所以它对 D_f 和 D_r 的输出相比于测试集有着较高的置信度, 所以熵值较低. 对于重

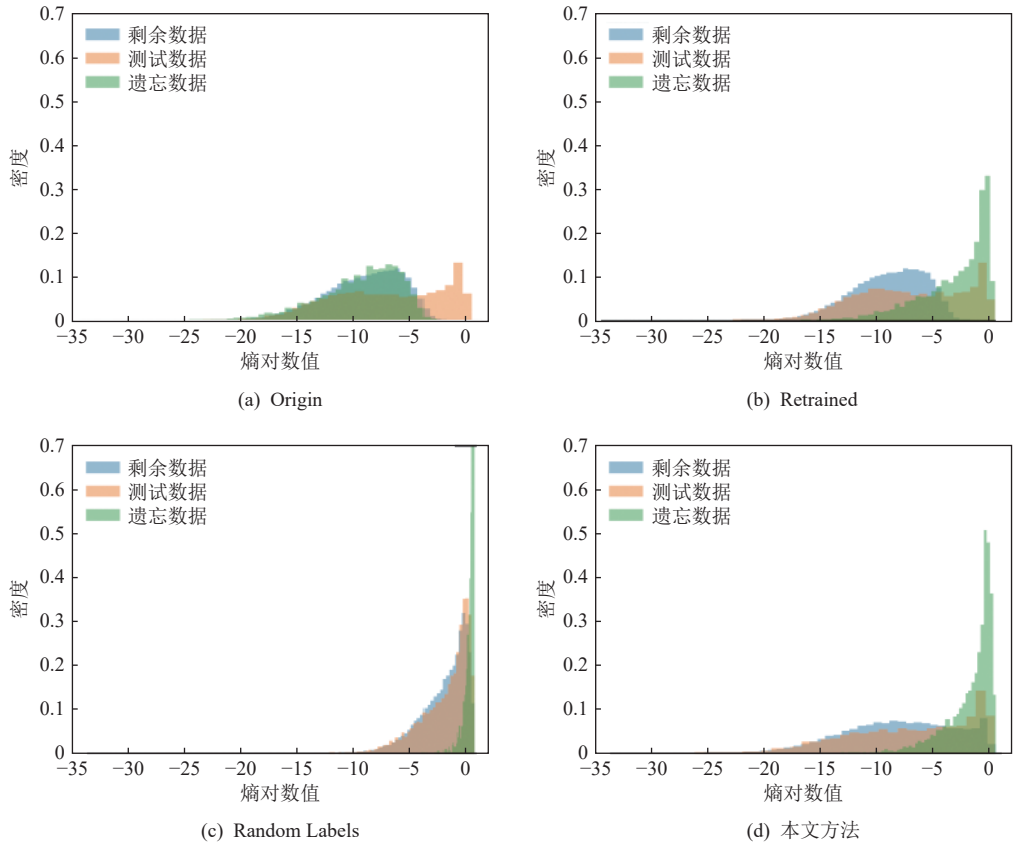


Fig. 6 Distribution of the model's output entropy on the retaining data, the forgetting data and the testing data

图 6 模型对剩余数据、遗忘数据、测试数据的输出熵分布

新训练模型，由于没有利用D_f训练，它对D_f的输出熵比D_r略高. 使用 random Labels 的遗忘模型对D_f的输出熵非常高，与 Retrained 差异很大. 它可能反而暴露更多关于D_f的信息，因为它只是从输出层面使 Origin 对D_f误分类，而没有消除D_f对模型的影响. 从图 6 可以看到，本文方法对D_f的输出熵与 Retrained 最接近，遗忘效果最好.

3.5 消融试验及分析

为了进一步研究本文方法中特征遗忘和自适应损失平衡的贡献，本节以 CIFAR10 数据集上的 ResNet18 模型为例，进行了消融实验. 我们在不改变“遗忘—恢复”架构的标准下，对“遗忘”阶段采用特征遗忘还是分类层面的遗忘、是否采用了自适应损失平衡等问题进行了分析，参与对比的方法为：

1)RT(Retrained). 重新训练模型(最优模型). 遗忘后模型与重新训练模型越接近越好.

2)LF(Logits-Forget). 在“遗忘”阶段，最小化随机初始化模型和遗忘模型的分类输出损失来优化模型，不采用自适应损失平衡(w_{fgt}和w_{rec}均为初始值，训练过程中保持不变).

3)LF+LB(Logits-Forget + Loss-Balance). 在“遗忘”阶段，最小化随机初始化模型和遗忘模型的分类输出损失来优化模型，采用自适应损失平衡.

4)FF(Feature-Forget). 在“遗忘”阶段，最小化随机初始化模型和遗忘模型的特征输出损失来优化模型，不采用自适应损失平衡(w_{fgt}和w_{rec}均为初始值，训练过程中保持不变).

5)FF+LB(Feature-Forget + Loss-Balance). 在“遗忘”阶段，最小化随机初始化模型和遗忘模型的特征输出损失来优化模型，采用自适应损失平衡.

利用上述方法在同一原始模型的基础上进行机器遗忘后，模型在剩余数据集上的准确率如图 7 所示，预测准确率均为 77.68%. 其中，所有方法得到的遗忘模型在D_f上的准确率均小于等于 1%. 可以看到，使模型在D_f上的准确率降到同样标准(1%)的条件下，将遗忘约束在特征层面相比于仅在分类层面实现遗忘的遗忘方法，能够更好地保护模型的泛化性能，如图 7 中的 LF, FF 以及 LF+LB, FF+LB 的对比所示，相比于 LF, FF 在遗忘 1 个类的数据时，模型在D_r上的准确率提高了 5 个百分点左右. 这说明特征遗忘可以更好地擦除模型中关于D_f的信息. 另一方面，对比 LF, LF+LB 以及 FF, FF+LB 可以发现，通过加入自适应损

失平衡，进一步提高了模型在D_r上的准确率（相比于FF，FF+LB在遗忘1个类的数据时，模型在D_r上的准确率提高了2个百分点左右）. 这主要是因为，初始时刻遗忘模型与随机初始化模型的特征输出差异较大，模型更倾向于对“遗忘”过程的优化，只进行了较少的“恢复”. 而自适应损失平衡可以更好地衡量“遗忘”和“恢复”任务对模型训练的贡献，以更加公平的方式使2个任务稳步训练. 同时，我们可以注意到，数据的不平衡性导致LF和FF在遗忘数据较少的情况下表现欠佳（例如，LF在遗忘1个类时，模型在D_r上的准确率仅为69.18%），随着遗忘数据与剩余数据的样本数量逐渐接近，LF和FF与采用自使用损失平衡的对比方法的差距总体上呈减小的趋势. 总的来说，本文方法FF+LB更好地保留了模型的泛化能力，与Retrained最接近.

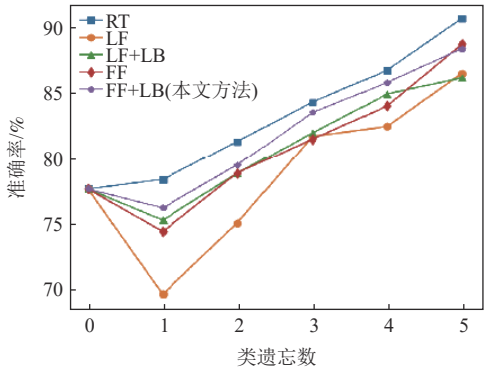


Fig. 7 Comparison of ablation studies on ResNet-18 model and CIFAR10 dataset

图7 基于ResNet18模型和CIFAR10数据集的消融实验对比

4 结 论

本文针对深度学习场景下的数据要素安全保护问题，提出了一个基于特征约束和自适应损失平衡的机器遗忘方法，以应对近似遗忘中难以兼顾遗忘效果和模型泛化性能的问题，支撑了人工智能时代数据要素安全中“被遗忘权”的实现. 本文采用了“遗忘—恢复”的两阶段遗忘框架，并将机器遗忘看作一个多任务优化问题. 首先，对于“遗忘”任务，我们考虑模型对数据特征层的学习，让原始模型学习随机初始化模型对于遗忘数据的特征输出，避免模型习得遗忘数据的信息. 其次，对遗忘模型进行微调来“恢复”模型在剩余数据上的性能. 同时，我们引入了自适应损失平衡来自动地权衡“遗忘”和“恢复”任务的训练，使2个任务的损失以较为平衡的方式稳步下降，避免单一任务的过度影响. 通过在准确率、时间开销、成员推理攻击等多个方面进行实验分析，本文方法与各种基线方法相比，取得了非常有竞争力的结果.

作者贡献声明：殷昱煜提出了写作思路和指导意见；吴广强负责实现并撰写论文；李尤慧子和高洪皓提出意见并修改论文；王鑫雨负责调研并辅助实验.

参 考 文 献

[1] Pulido-Gaytan L B, Tchernykh A, Cortés-Mendoza J M, et al. A survey on privacy-preserving machine learning with fully homomorphic encryption[C]//Proc of Latin American High Performance Computing Conf. Cham: Springer International Publishing, 2020: 115−129

[2] Liu Yi, Xu Lei, Yuan Xingliang, et al. The right to be forgotten in federated learning: An efficient realization with rapid retraining[C]// Proc of IEEE Conf on Computer Communications (IEEE INFOCOM 2022). Piscataway, NJ: IEEE, 2022: 1749−1758

[3] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[C/OL]//Proc of the Int Conf on Learning Representations. 2015 [2024-05-28]. https://arxiv.org/abs/1409.1556

[4] Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[J]. Advances in Neural Information Processing Systems, 2014, 27: 2672−2680

[5] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in Neural Information Processing Systems, Cambridge, MA: MIT Press, 2017: 5998−6008

[6] Fredrikson M, Jha S, Ristenpart T. Model inversion attacks that exploit confidence information and basic countermeasures[C]//Proc of the 22nd ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2015: 1322−1333

[7] Voigt P, Von dem Bussche A. The EU General Data Protection Regulation (GDPR)[EB/OL]. https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32016R0679

[8] Bourtoule L, Chandrasekaran V, Choquette-Choo C A, et al. Machine unlearning[C]//Proc of 2021 IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE, 2021: 141−159

[9] Yan Haonan, Li Xiaoguang, Guo Ziyao, et al. ARCANE: An efficient architecture for exact machine unlearning[C]//Proc of the 31st Int Joint Conf on Artificial Intelligence. Menlo Park, CA: AAAI, 2022: 4006−4013

[10] Dukler Y, Bowman B, Achille A, et al. SAFE: Machine unlearning with shard graphs[C]//Proc of the IEEE/CVF Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2023: 17108−17118

[11] Golatkar A, Achille A, Soatto S. Eternal sunshine of the spotless net: Selective forgetting in deep networks[C]//Proc of the IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2020: 9304−9312

[12] Huang H, Ma X, Erfani S M, et al. Unlearnable examples: Making

personal data unexploitable[C/OL]//Proc of the Int Conf on Learning Representations. 2021 [2024-01-17]. https://arxiv.org/abs/2101.04898

[13] Tarun A K, Chundawat V S, Mandal M, et al. Fast yet effective machine unlearning[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023. Doi: 10.1109/TNNLS.2023.3266233, 2023

[14] Kim J, Woo S S. Efficient two-stage model retraining for machine unlearning[C]//Proc of the IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2022: 4361−4369

[15] Chundawat V S, Tarun A K, Mandal M, et al. Can bad teaching induce forgetting? unlearning in deep networks using an incompetent teacher[C]//Proc of the AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2023, 37(6): 7210−7217

[16] Chen Min, Gao Weizhuo, Liu Gaoyang, et al. Boundary unlearning: Rapid forgetting of deep networks via shifting the decision boundary[C]//Proc of the IEEE/CVF Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2023: 7766−7775

[17] Vandenhende S, Georgoulis S, Van Gansbeke W, et al. Multi-task learning for dense prediction tasks: A survey[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 44(7): 3614−3633

[18] Zhao Shiji, Yu Jie, Sun Zhenlong, et al. Enhanced accuracy and robustness via multi-teacher adversarial distillation[C]//Proc of European Conf on Computer Vision. Cham: Springer, 2022: 585−602

[19] Kirkpatrick J, Pascanu R, Rabinowitz N, et al. Overcoming catastrophic forgetting in neural networks[J]. Proceedings of the National Academy of Sciences, 2017, 114(13): 3521−3526

[20] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network[J]. Computer Science, 2015, 14(7): 38−39

[21] Zhang Xulong, Wang Jianzong, Cheng Ning, et al. Machine unlearning methodology based on stochastic teacher network[C]//Proc of Int Conf on Advanced Data Mining and Applications. Cham: Springer, 2023: 250−261

[22] Guo Husheng, Zhang Yang, Wang Wenjian. Two-stage adaptive ensemble learning method for different types of concept drift[J]. Journal of Computer Research and Development, 2024, 61(7): 1799−1811(in Chinese)
(郭虎升, 张洋, 王文剑. 面向不同类型概念漂移的两阶段自适应集成学习方法[J]. 计算机研究与发展, 2024, 61(7): 1799−1811)

[23] Foster J, Schoepf S, Brintrup A. Fast machine unlearning without retraining through selective synaptic dampening[C]//Proc of the AAAI Conf on Artificial Intelligence. Palo Alto, CA: AAAI, 2024, 38(11): 12043−12051

[24] Aich A. Elastic weight consolidation (EWC): Nuts and bolts[J]. arXiv preprint, arXiv: 2105.04093, 2021

[25] Cotogni M, Bonato J, Sabetta L, et al. DUCK: Distance-based unlearning via centroid kinematics[J]. arXiv preprint, arXiv: 2312.02052, 2023

[26] Paszke A, Gross S, Massa F, et al. Pytorch: An imperative style, high-performance deep learning library[C]//Advances in Neural Information Processing Systems Cambridge, MA: MIT, 2019: 8024−8035

[27] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images[R]. Toronto: University of Toronto, 2009

[28] LeCun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278−2324

[29] Springenberg J T, Dosovitskiy A, Brox T, et al. Striving for simplicity: The all convolutional net[C/OL]//Proc of the Int Conf on Learning Representations. 2015 [2024-02-02]. https://arxiv.org/abs/1412.6806

[30] He Kaiming, Zhang Xiangyu, Ren Shaoqing, et al. Deep residual learning for image recognition[C]//Proc of the IEEE Conf on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2016: 770−778

[31] Shokri R, Stronati M, Song C, et al. Membership inference attacks against machine learning models[C]//Proc of 2017 IEEE Symp on Security and Privacy (SP). Piscataway, NJ: IEEE, 2017: 3−18

[32] Li Wenbin, Xiong Yakun, Fan Zhichen, et al. Advances and trends of continual learning[J]. Journal of Computer Research and Development, 2024, 61(6): 1476−1496 (in Chinese)
(李文斌, 熊亚锟, 范祉辰, 等. 持续学习的研究进展与趋势[J]. 计算机研究与发展, 2024, 61(6): 1476−1496)

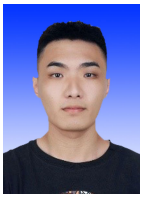
[33] Goodfellow I J, Shlens J, Szegedy C. Explaining and harnessing adversarial examples[C/OL]//Proc of the Int Conf on Learning Representations. 2015 [2024-02-02]. https://arxiv.org/abs/ 1412.6572

[34] Selvaraju R R, Cogswell M, Das A, et al. Grad-CAM: Visual explanations from deep networks via gradient-based localization[C]//Proc of the IEEE Int Conf on Computer Vision. Piscataway, NJ: IEEE, 2017: 618−626



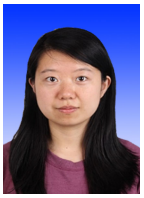
Yin Yuyu, born in 1980. PhD, professor. Member of CCF. His main research interests include service computing, edge computing, and business process management.

殷昱煜, 1980 年生. 博士, 教授. CCF 会员. 主要研究方向为服务计算、边缘计算、业务流程管理.



Wu Guangqiang, born in 1999. Master. His main research interests include machine unlearning, deep learning, and incremental learning.

吴广强, 1999 年生. 硕士. 主要研究方向为机器遗忘、深度学习、增量学习.



Li Youhuizi, born in 1989. PhD, associate professor. Member of CCF. Her main research interests include edge computing, privacy security, and mobile edge computing.

李尤慧子, 1989 年生. 博士, 副教授. CCF 会员. 主要研究方向为边缘计算、隐私安全、移动边缘计算.



Wang Xinyu, born in 2000. Master. His main research interests include edge computing, deep learning, and natural language processing.

王鑫雨, 2000 年生. 硕士. 主要研究方向为边缘计算、深度学习、自然语言处理.



Gao Honghao, born in 1985. PhD, associate professor. Member of CCF. His main research interests include software formal verification, service collaborative computing, wireless networks and industrial Internet of things, and intelligent medical image processing.

高洪皓, 1985 年生. 博士, 副教授. CCF 会员. 主要研究方向为软件形式化验证、服务协同计算、无线网络和工业物联网、智能医学影像处理.