Maintaining Randomised Trees for Low-Latency Machine Unlearning

- ABSTRACT
  - 机器学习是从用户历史数据中学习的系统软件
  - 法律“一般数据保护条例”（GDPR）中颁布“被遗忘的权利”，其要求处理个人数据的组织根据请求删除用户数据。
    - 该规定不仅要求从数据库中删除用户数据，还要求从存储数据中学习的ML模型删除用户的历史数据。
  - 因此，作者认为ML应用程序应该为用户提供及时从经过训练的模型中删除学习数据的机会。
  - 作者将探讨在现实世界部署的约束下，这种取消学习的速度有多快，并介绍低延迟机器遗忘学习存在的问题：在移除一小部分训练样本而无需重新训练的情况下，保持已部署的ML模型。
  - 作者提出了HedgeCut，这是一种基于随机决策树集合的分类模型，旨在以低延迟回答遗忘请求。
    - 详细介绍了如何使用矢量化算子有效地实现HedgeCut，以进行决策树学习。
- INTRODUCTION
  - 最近的法律，如“被遗忘权”（《通用数据保护条例》（GDPR）第 17 条）要求处理个人数据的公司和机构应要求删除用户数据：“数据主体应有权 [...] 在数据主体撤回同意的情况下，不得无故拖延地删除与他或她有关的个人数据 [...]”。
  - 最近的研究认为，仅仅从数据库等原始数据存储中删除个人用户数据是不够的，并且已经针对存储数据进行了培训的机器学习模型也属于该法规的范畴。
  - 当前问题
    - 关系数据库为数据的物化视图提供事务性删除和相应更新，但是对于从数据派生的 ML 模型不存在这种自动删除机制。
    - 近年来，机器学习社区一直在机器忘却学习的保护伞下研究这个问题。
      - 给定一个模型、它的训练数据和一组要取消学习的用户数据，他们提出了加速模型再训练的有效方法。
      - 然而，这些方法缺乏数据管理的观点，因为它们忽略了现实世界 ML 应用程序中部署管道的复杂性所施加的约束。
      - ML 模型部署在服务系统中，可以有效地以低延迟响应在线预测请求，但为了更新模型，必须执行必须启动基础设施、重新训练模型和运行昂贵的评估工作负载的重量级管道。
      - 在整个过程中，仅仅为了忘却一个培训示例，在经济上和操作上都是没有意义的。
    - 然而，在理想情况下，我们仍然希望能够在定期调动之间立即忘却单一的例子。

    - 为此，我们必须能够完全绕过繁重的重新部署步骤。
    - 我们可以通过点查询检索要删除的用户的数据，并且希望能够向服务系统发出删除请求，以简单地更新已部署的模型（而不必训练和部署新模型）。
- 在实际部署中使用Spark MLlib时，重新训练和重新部署模型的高昂成本：
    - 在开始训练之前，我们需要在云中配置机器；
    - 接下来，我们需要在集群上启动 Spark，并将模型的训练数据从其原始位置（通常是安全的分布式文件系统）读取到集群的聚合主内存中；
    - 现在，我们可以根据更新的训练数据从头开始重新训练模型；
    - 之后，通常会运行一组额外的健全性测试，例如，检查发生变化的预测，或在回测场景中将新模型的性能与当前模型的性能进行比较；
    - 最后，我们启动重新部署到服务系统，通常使用“金丝雀”和“回滚”步骤，以便能够对有缺陷的模型版本做出反应； 在许多情况下，这种重新部署可能需要我们启动新的服务机器，以便可以自动重新路由流量。
- 总结
    - 在整个过程中，仅仅为了忘却一个培训示例，在经济上和操作上都是没有意义的。
    - 然而，在理想情况下，我们仍然希望能够在定期调动之间立即忘却单一的例子。
    - 为此，我们必须能够完全绕过繁重的重新部署步骤。
    - 在实际部署中，执行完整的模型再培训效率低下且成本高昂，因为再培训和重新部署需要执行复杂的重型数据管道。