中国科学院软件研究所学术年会'2021
暨计算机科学国家重点实验室开放周

工具系统

数据库与深度学习系统间数据传输工具

研究团队：吴铭钞、秦政、许利杰、王伟
软件工程技术研究开发中心

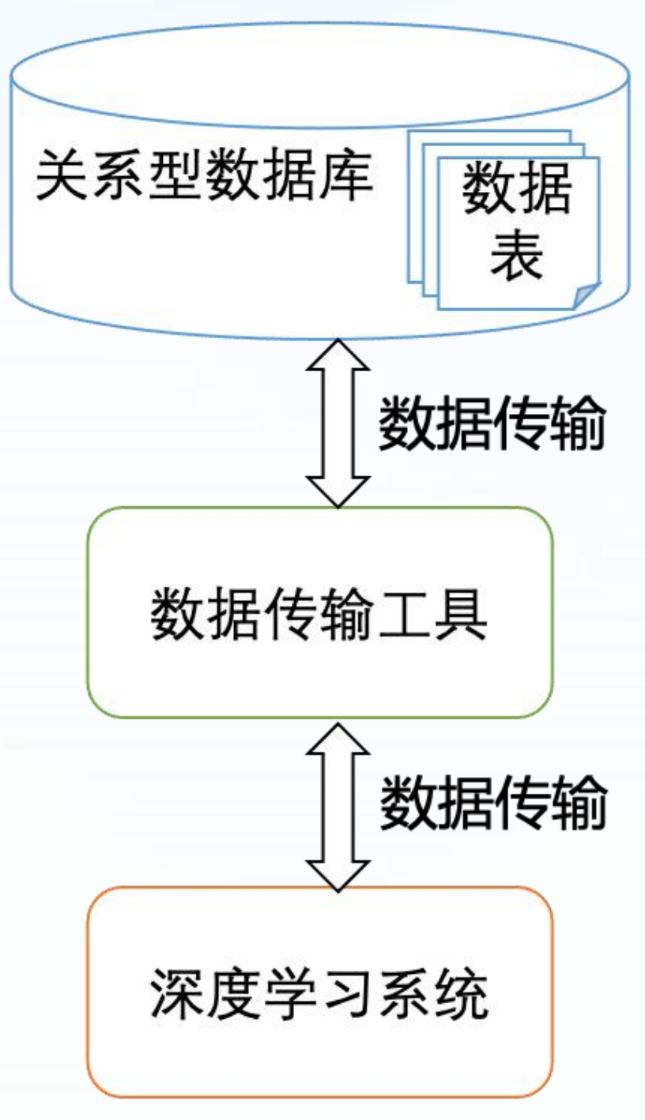
xulijie09@otcaix.iscas.ac.cn wangwei@otcaix.iscas.ac.cn

工作介绍

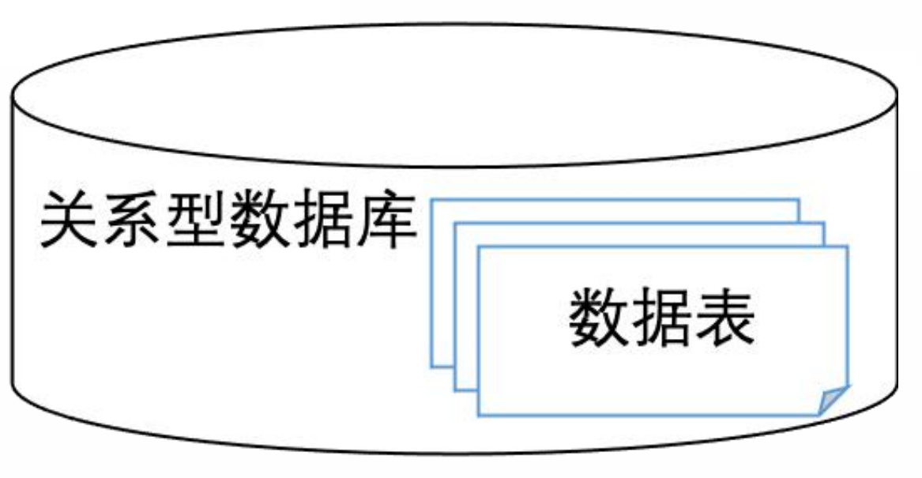
场景：关系型数据库中存储着大量的数据，是深度学习任务的天然数据源，而深度学习任务则高度依赖各类深度学习系统。因此，需要一种关系型数据库和深度学习系统间的数据传输工具。

挑战：现有的DB4AI工作主要专注于传统的统计机器学习，故本课题关注设计并实现关系型数据库与深度学习系统间高效、易用的数据传输工具。

工作内容：本课题对现有问题进行分析，设计并实现了国产关系型数据库DM8与深度学习系统 PyTorch 间的数据传输工具，使用了本地缓存及并行化预取等优化策略。实验结果表明，本工具的数据传输结果正确，数据传输效率高。

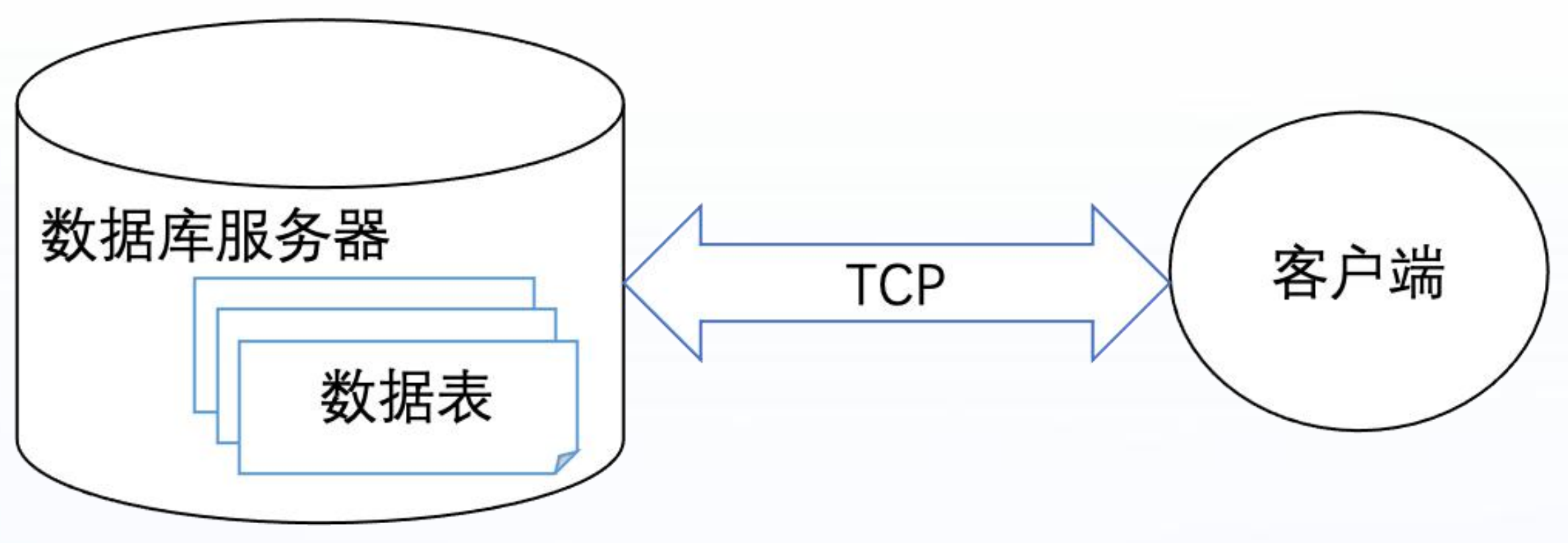


问题分析

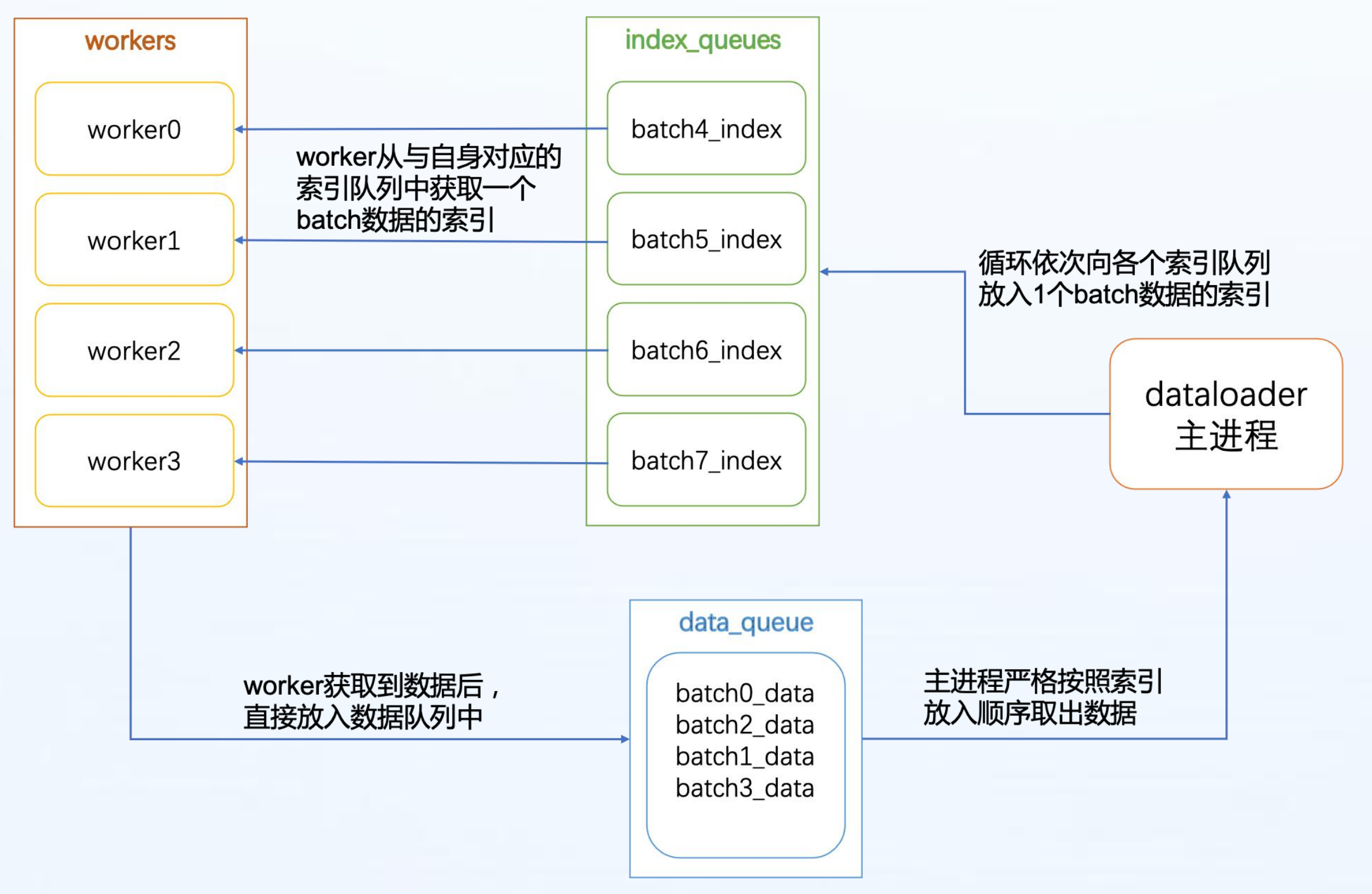


数据存储：关系型数据库中，原始数据一般存储在多张数据表中，存在冗余、格式不规范等问题，难以直接用于深度学习。

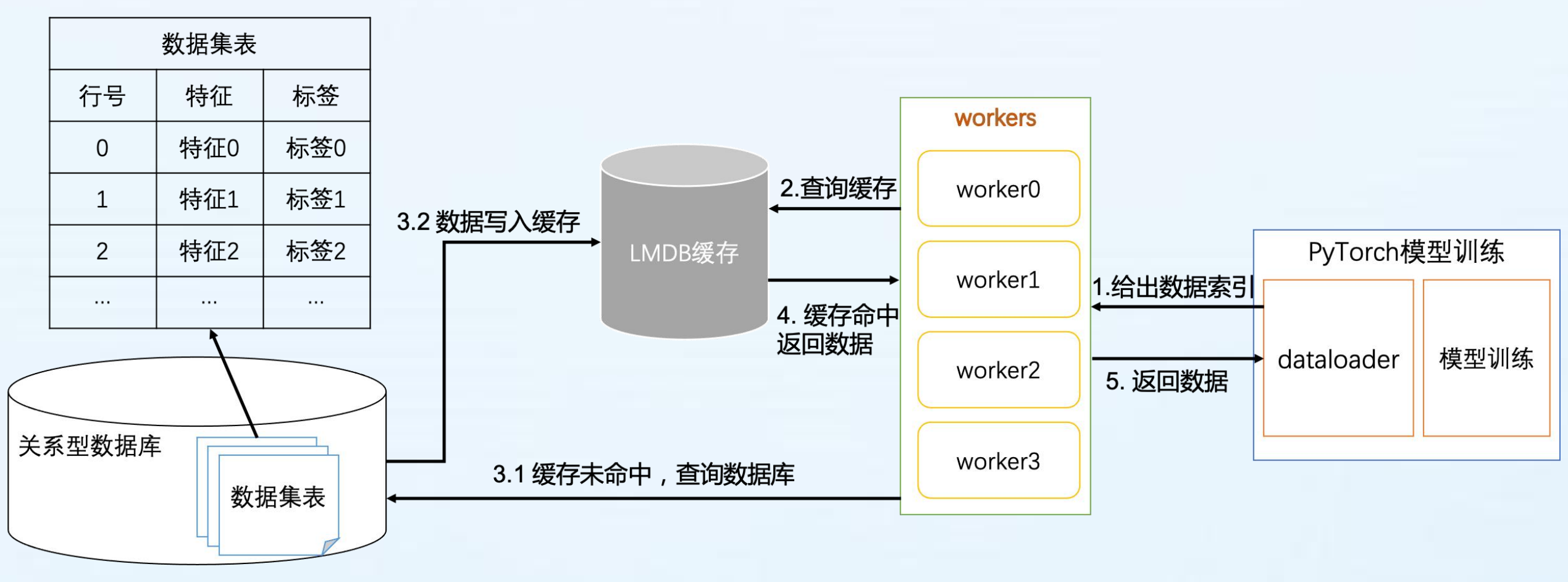
数据传输：数据库服务器和客户端之间使用TCP协议，通过网络进行数据传输，这一过程速度慢、受网络条件影响大。



数据加载：为了提升数据加载速度，应尽可能使用并行化方法。PyTorch提供了成熟的多进程并行化数据加载接口，用户需要自己实现良好的并行化代码。

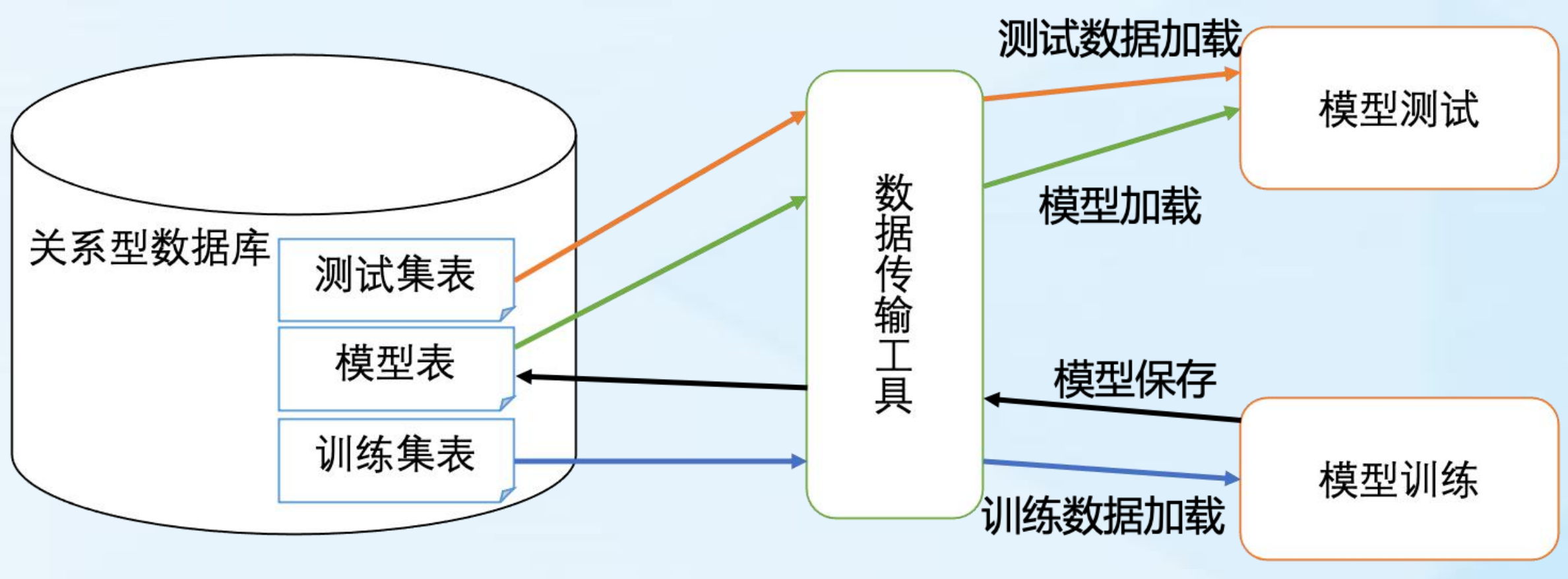


工具实现



本项目使用Python语言，基于DM8的Python驱动以及PyTorch的相关接口，实现了数据传输工具，支持：

- 利用多进程并行化地从数据库读取数据
- 使用LMDB缓存实现高性能的本地数据缓存
- 利用LMDB缓存配合实现预取



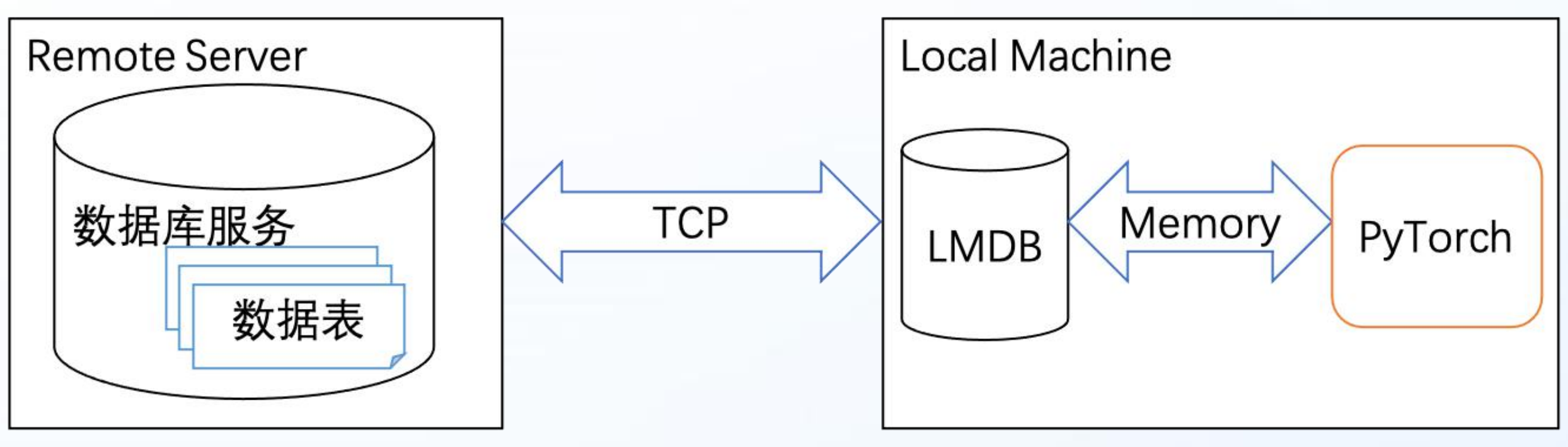
使用工具，可从数据库中加载数据集，进行模型训练，然后将模型训练的结果导出并保存到数据库中，之后可由数据库恢复模型，用于测试和预测。

工具设计

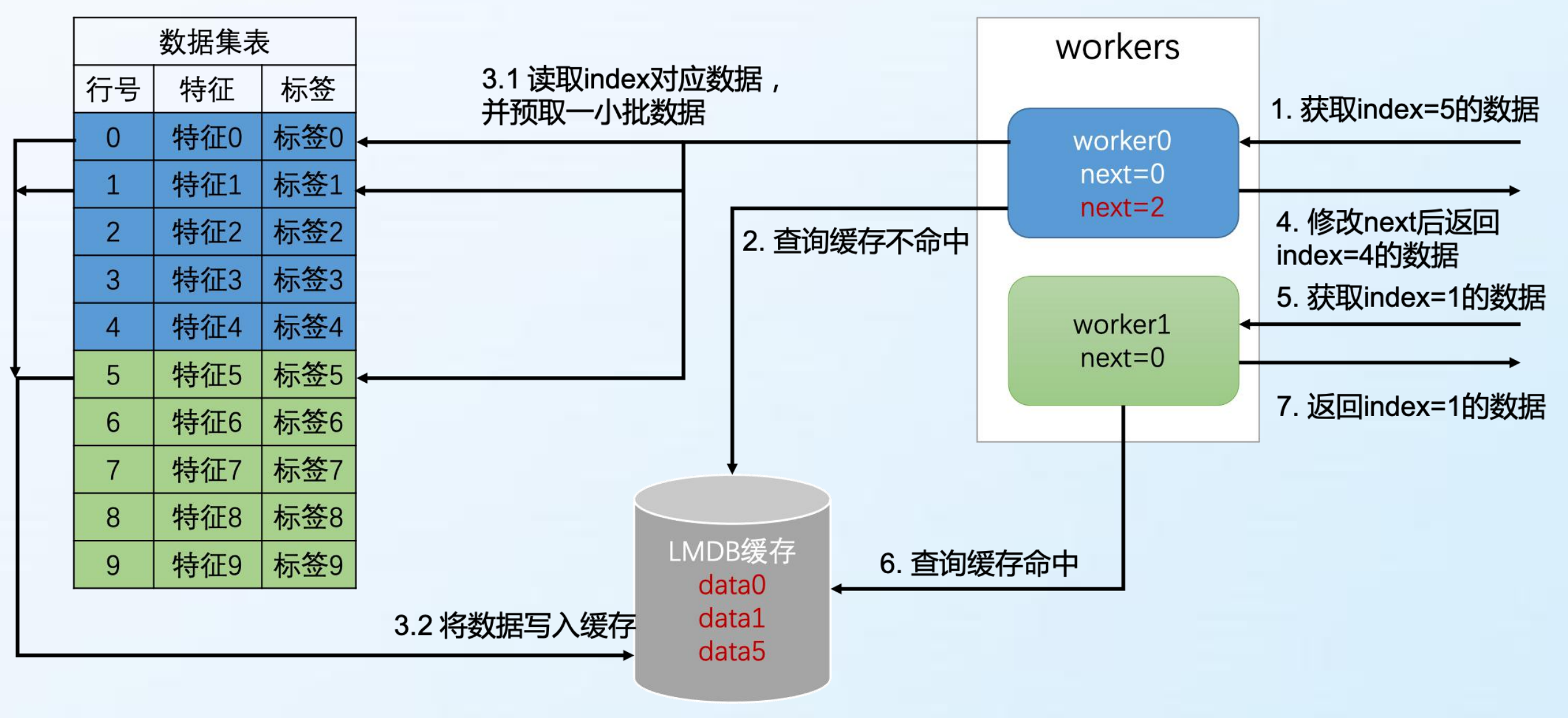
数据集表		
行号	特征	标签
0	特征0	标签0
1	特征1	标签1
2	特征2	标签2
…	…	…

数据集表：为减少冗余数据，仅存储必要的数据特征和标签；为保证查询高效性，数据集表使用一个递增行号列作为主键。

LMDB缓存：为减少网络传输数据带来的开销，在深度学习任务的第一个epoch，将数据缓存到LMDB（一个高性能的内存映射K/V数据库）中，以便后续使用。

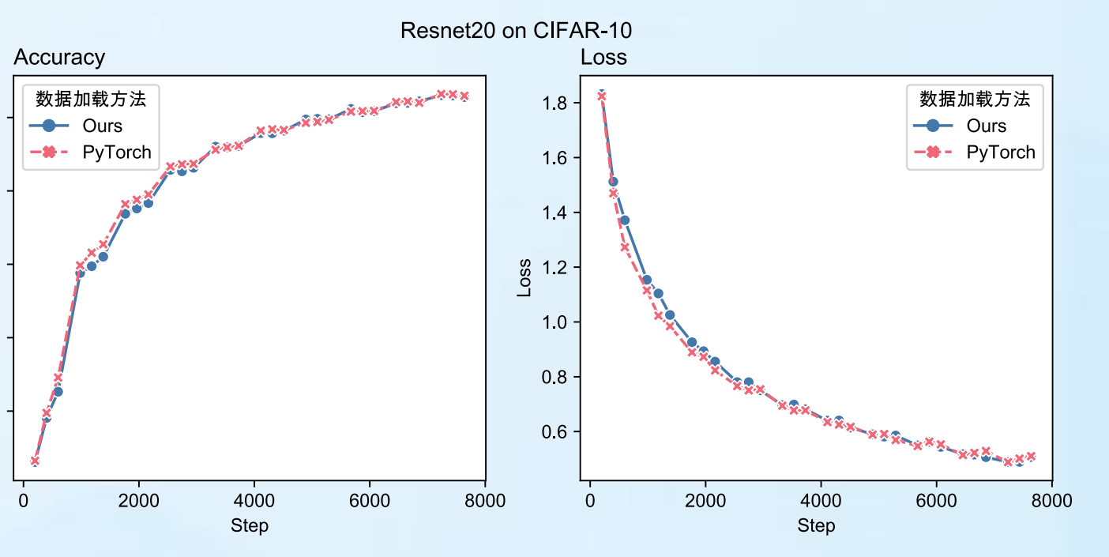


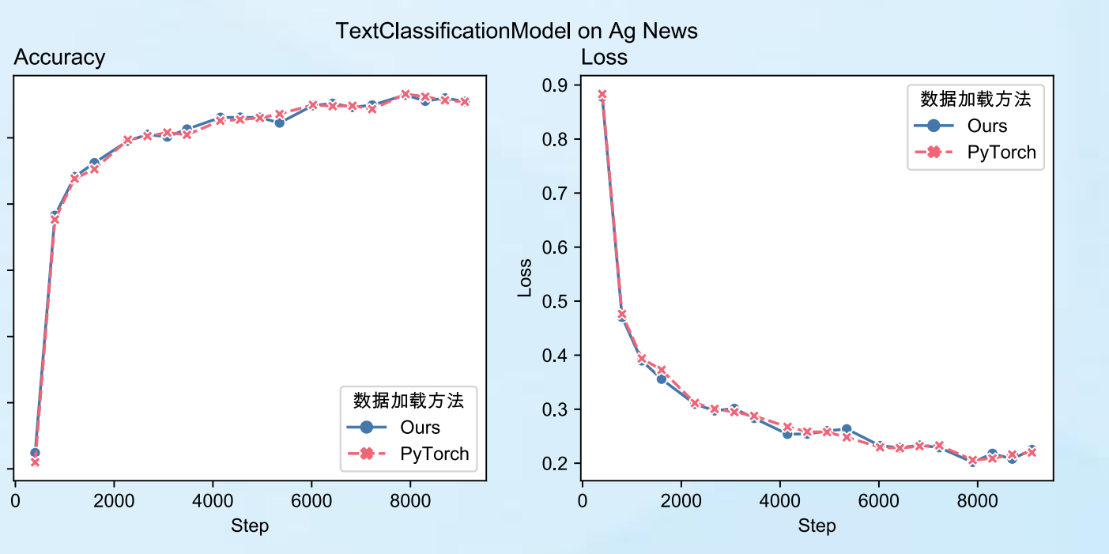
多进程预取：用户可指定若干个进程并行加载数据。将数据集表均分为若干段，每个进程负责读取其中一段，配合LMDB，每次从数据库服务器查询数据时，进行预取，尽可能减少数据库查询次数。

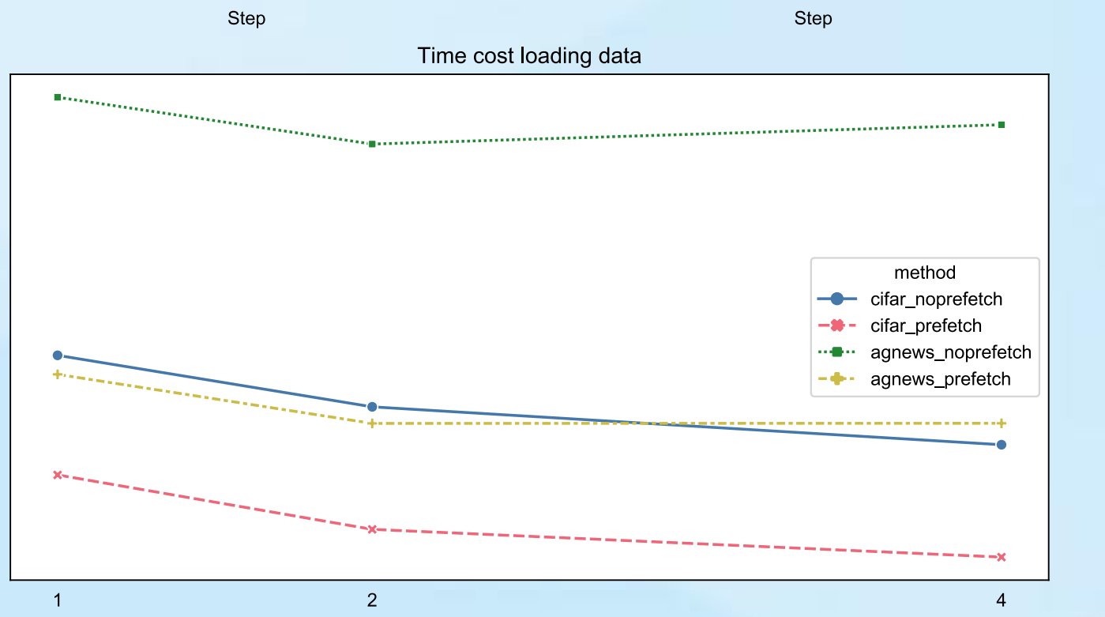


实验结果

利用本工具，使用CIFAR-10训练resnet20，使用Ag News训练TextClassifacationModel，与PyTorch自带数据集进行对比。







数据加载正确性：左上侧4图分别显示了使用本工具从数据库加载数据以及使用PyTorch自带数据集训练模型时的准确率、损失值曲线，可见两者基本一致。

数据加载效率：左下侧图显示了在是/否使用预取以及不同并行度的情况下，两个数据集的加载速度。可见多进程并行及预取的加速效果明显。

实验结论：

- 本数据传输工具的数据加载正确性无误；
- 并行化预取的策略有效加快了数据加载效率。