**0 Survey and Tutorial**

**0.1 Survey**

## 0.1.1 Database Meets Deep Learning: Challenges and Opportunities（SIGMOD Record-2016）

（1）综述。主要讲数据库领域和深度学习领域交叉的研究问题；从数据库角度讨论深度学习系统可能的改进，以及数据库应用程序如何从深度学习技术中受益；

（2）在深度学习中，涉及较多的代价昂贵的线性运算，包括矩阵乘法；为了加速运算，将数据依赖性不强的操作放在执行机器上并行计算，如CUDA流和CPU线程等；也可以调整读写顺序动态完成；数据库在优化事务执行时和查询计划时，也可以用类似这种解决方案**（我的理解是在事务执行时使用多个机器将数据相关性不强的操作并行执行、在制定查询计划时构造包含执行器和内存等的计算资源成本模型得到最优查询计划）**；

（3）数据库系统通过日志记录和检查点保证持久性，深度学习系统的训练主要基于检查点文件从崩溃中恢复训练，但是频繁的检查点会带来巨大的开销；与数据库系统在事务中强制执行严格的一致性相比，深度学习训练系统使用的SGD算法能够容忍一点过的不一致性，因此日志记录不是必须的；

（4）在数据库上理解自然语言的语义是一项挑战，可以通过应用RNN模型来解析自然语言查询生成SQL查询，并使用现有的数据库方法对其进行改进。

（5）当前大多数数据库系统使用复杂的启发式和成本模型来生成查询计划，参数SQL查询模板的每个查询计划都有一个最优性区域，只要SQL查询在这个区域内，最优查询计划就不会改变。即查询计划对于输入的参数微小变化并不敏感，因此可以训练一个查询规划器，它从一组SQL查询和最优计划中学习；

（6）总结：主要讨论数据库优化系统性能的技术，在内存优化和并行上讨论一下可能的改进；

**0.2 Tutorial**

## 0.2.1 From auto-tuning one size fits all to self-designed and learned data-intensive systems.（SIGMOD-2019)

（1）主要在数据库系统层面讨论当前的创新点和优化改进可能的机会。

（2）讨论在数据库系统设计中的总体问题设置。

讨论过去几十年的主要研究和行业趋势，包含自动调优（在线和离线）、自适应索引、广义索引、模块化系统等。

讨论在上述方面，与之前的解决方案相比我们有哪些新的解决机会，以及在实际的问题中我们如何应用在许多类型的数据密集型应用。

（3）离线索引；数据库系统自动调优的最早方法，所有的数据库产品都会提供自动调优工具来自动化数据库物理设计，该工具分析给定的工作负载，并建议适当的物理设计。离线索引需要数据库管理员（DBA）的大量参与；

（4）在线索引；通过在线索引，系统持续监控工作负载，并定期重新评估物理设计；需要大量的时间和资源，主要适用于查询模式更改不太频繁的中等动态工作负载；

（5）自适应索引和布局；一种轻量级的方法，解决了动态工作负载的离线和在线索引的限制，通过部分的和增量的构建或优化索引来响应工作负载的变化，期间不需要DBA或者脱机处理；自适应索引对每个查询做出轻量级的响应，索引越精细性能越好。自动调优和自适应没有从根本上改变系统的属性。

（6）模块化系统；使用较小的努力轻松的添加数据类型，或者即插即用的功能和具有该井接口的整个系统组件。针对系统构建和数据密集型系统，模块化是一个非常有前途的方向，但目前只有“大型”组件的系统，并不真正允许系统彻底改变其行为；

（7）广义树索引（GiST）；目标是使扩展用于索引的数据结构变得容易，并以最小的努力将他们定制为特定的问题和数据；它是一个抽象的索引定义的模板；拥有对并发的支持、更通用的API、改进的性能和对生成的索引的选择性估计；

（8）自行设计系统。自行设计的系统利用设计空间自动生成适用于目标工作负载和硬件的设计。根据学习成本模型可以学习基本访问模式的成本，然后给定数据结构规范的复杂算法的成本，这些成本可以被机器学习算法使用，通过迭代生成的数据结构规范来标记设计，并计算奖励，以确定下一步尝试哪种规范。

（9）学习系统，在该系统中，传统的核心数据系统组件被模型所取代；例如学习索引模型替换数据结构的索引部分；

（10）可以扩展的研究：

对于自行设计系统：高效搜索最优设计，快速高效地生成目标设计代码，易于扩展以扩展支持的设计空间；

对于学习系统：高效存储模型，使用现代硬件高效执行复杂模型，结果的鲁棒性和可解释性；