Qlib:面向人工智能的量化投资平台

杨晓、刘维青、周东、边江、刘铁燕微软研究院
{肖。杨,维情。刘、周。扁,Tie-Yan.Liu} @microsoft.com

摘要

量化投资的目标是在一系列金融工具的连续交易期内实现收益最大化和风险最小化。最近，受人工智能技术在量化投资领域产生显著创新方面的快速发展和巨大潜力的启发，越来越多的人采用人工智能驱动的工作流程进行量化研究和实际投资。人工智能技术在丰富量化投资方法论的同时，也对量化投资体系提出了新的挑战。特别是，量化投资的新学习范式要求基础设施升级，以适应更新的工作流程;此外，人工智能技术的数据驱动性质确实要求基础设施具有更强大的性能;此外，应用人工智能技术来解决金融场景中的不同任务也存在一些独特的挑战。为了应对这些挑战，弥合人工智能技术和量化投资之间的差距，我们设计和开发了 Qlib，旨在实现人工智能技术在量化投资中的潜力，增强研究能力，创造价值。

1 介绍

量化投资是目前最热门的研究领域之一，吸引了众多学术界和金融界的优秀人才。近几十年来，通过对量化方法的不断优化，整个专业投资者群体总结出了一套完善但不完善的量化研究工作流程。最近，新兴的 AI 技术在这一研究领域开启了一种新的趋势。随着人们越来越重视挖掘 AI 在量化投资中的巨大潜力，AI 技术已被量化研究者广泛应用于实际投资中。

AI 技术在丰富量化投资方法论的同时，也从多重向量化投资体系提出了新的挑战视角。首先，由 AI 技术的灵活性引起的量化投资工作流程的技术革命，往往需要新的支持性基础设施。例如，传统的量化投资通常将整个工作流程拆分为几个子任务，包括股票趋势预测、投资组合优化等，而 AI 技术使得建立端到端的解决方案，直接生成最终的投资组合成为可能。为了支持这种端到端解决方案，由于其数据驱动的性质，有必要对当前的基础设施进行升级。

同时，人工智能技术必须处理一些新场景中的独特问题，这既需要大量的金融领域知识，也需要丰富的数据科学经验。在没有任何领域适应的情况下，将解决方案应用于定量研究任务很少奏效。这样的情况导致迫切需要一个平台来适应 AI 时代这样的现代量化研究工作流程，并为 AI 技术在金融场景中的应用提供指导。

因此，我们提出了一个新的面向 ai 的量化投资平台，叫做 Qlib[1]。它旨在协助探索人工智能技术在量化投资中的巨大潜力的研究工作，并使量化研究人员能够在人工智能驱动的量化投资上创造更显著的价值。具体而言，Qlib 面向人工智能的框架旨在适应基于人工智能的解决方案。此外，它还提供了专用于量化投资场景的高性能基础设施，这使得许多 AI 研究课题成为可能。此外，Qlib 集成了一批专为量化投资场景下机器学习设计的工具，让用户在充分利用 AI 技术方面受益。

最后，我们通过比较量化投资中一个典型任务的几种解决方案，演示了一些用例，并评估了 Qlib 基础设施的性能。结果表明，Qlib 专用于量化投资的基础设施在此任务上优于大多数现有解决方案。

2 背景及相关工作

在本节中，我们将首先演示现代定量研究者在应用时的主要实际问题

[1] 代码可在 https://github.com/microsoft/qlib 获取

人工智能技术在量化投资中的重要性，这推动了 Qlib 的诞生。之后，我们将简要介绍相关工作。

2.1 实际问题

量化研究工作流革命

在传统的投资研究工作流中，研究人员往往根据几个因素(因素类似于机器学习中的特征)和基本的金融数据，通过线性模型[Petkova, 2006]或手动设计规则[Murphy, 1999]来开发交易信号。然后，遵循交易策略(通常是 Barra[Sheikh, 1996])来生成目标投资组合。最后，研究人员通过回测函数来评估交易信号和投资组合。

随着 AI 技术的兴起，它对传统的量化投资发起了一场技术革命。传统的量化研究工作流程过于原始，无法容纳如此灵活的技术。为了更直观地展示差异，我们将展示一个典型的基于 AI 技术的现代研究工作流。它从一个具有大量特征(通常超过数百个维度)的数据集开始。手动设计这么多的特征需要花费大量的时间。利用机器学习算法自动生成这样的特征是很常见的[Potvin *et al.*， 2004;Neely *et al.*， 1997;Allen and Karjalainen, 1999;Kakushadze, 2016]。生成数据[Feng *et al.*， 2019]是构建数据集的另一种选择。基于不同的数据集，研究人员提供了数百种机器学习方法来挖掘交易信号[Sezer *et al.*， 2019]。研究人员可以根据这样的交易信号生成目标投资组合。但这样的工作流程并不是唯一的选择。RL(强化学习)不是将任务划分为几个阶段，而是直接提供了从数据到最终交易动作的端到端解决方案[Deng *et al.*， 2016]。RL 通过与环境的交互来优化交易策略，环境是金融场景中的交易模拟器。RL 需要一个响应式模拟器，而不是传统研究工作流程中的回测功能。而且，大多数 AI 算法都有复杂的超参数，需要仔细调优。

AI 技术如此灵活，已经超出了为传统方法学设计的现有工具的范围。从零开始构建基于 AI 技术的研究工作流程需要花费大量时间。

对基础设施的高性能要求高

随着 AI 技术的出现，对基础设施的要求也发生了变化。这样一种数据驱动的方法可以利用大量的数据。在高频交易场景下，数据量可以达到 TB 量级。此外，从基本的价格和交易量数据中衍生出数千个新特征(例如 Alpha101 [Kakushadze, 2016])是非常常见的，而这些数据总共只包含 5 个维度。一些研究人员甚至试图通过搜索表达式来创建新的因子或特征[Allen and Karjalainen, 1999;Neely *et al.*, 1997;Potvin *et al.*， 2004]。如此繁重的数据处理工作让研究人员负担过重，甚至做了一些研究主题是不可能的。这样的情况对基础设施提出了更加严格的性能要求。

应用机器学习解决方案的障碍

财务数据和任务有其独特性和挑战。将机器学习解决方案应用于量化研究任务而不进行任何调整很少奏效。由于金融数据的 SNR(信噪比)极低，在金融市场中建立成功的数据驱动策略是非常困难的。大多数机器学习算法都是数据驱动的，必须要处理这样的困难。如果不仔细处理细节，机器学习模型很难实现令人满意的性能。即使是一个小错误也会让模型过度拟合噪声，而不是学习有效的模式。正确处理细节需要大量的金融行业领域知识。此外，典型的目标，比如年化回报率，往往是不可微分的，这使得直接为机器学习方法训练模型变得困难。用适当的监督目标定义一个合理的任务，对于金融数据建模是非常重要的。这样的障碍让相当多对金融行业没有太多领域知识的数据科学家望而却步。

构建机器学习应用程序的另一个必要步骤是超参数优化。不同的机器学习算法有不同的超参数搜索空间，每个超参数搜索空间都有多个维度，具有不同的含义和优先级。一些量化研究人员来自传统金融行业，对机器学习的知识并不多。如此巨大的学习成本，让很多用户无法充分发挥机器学习的最大价值。

2.2 相关工作

在金融行业，一种投资策略的收益随着投资者的增多而降低。因此，金融从业者，尤其是量化研究者，从来不热衷于分享自己的算法和工具。OLPS [Li *et al.*， 2016]是第一个用于投资组合选择的开源工具箱。它由一系列由机器学习算法驱动的经典策略组成，作为基准和工具包，以促进新的学习方法的开发。这个工具箱只支持 Matlab 和 Octave，与当前科学主流语言 Python 并不兼容，因此对现代机器学习算法并不友好。它的框架相当简单，而基于 AI 技术的现代定量研究工作流要复杂得多。近年来出现了其他量化工具。QuantLib [Firth, 2004]只关注现代定量研究工作流程的一部分。QUANTAXIS[2] 更多地关注 IT 基础设施，而不是研究工作流程。Quantopian 发布了一系列开源工具 1)Alphalens:用于预测(alpha)股票因素性能分析的 Python 库 2)Zipline:用于回测的事件驱动系统 3)Pyfolio:用于金融投资组合性能和风险分析的 Python 库。它们都只专注于交易信号或投资组合的分析。

[2] https://github.com/QUANTAXIS/QUANTAXIS

总的来说，Qlib 是第一个适应人工智能时代现代定量研究人员工作流程的开源平台。它旨在使每一位量化研究人员都能意识到 AI 技术在量化投资中的巨大潜力。

3 以 ai 为导向的量化投资平台

3.1 总体设计

在与具有多年金融市场实践经验的量化研究人员的合作中，我们遇到了上述所有问题，并探索了各种解决方案。在当前环境的激励下，我们实施 Qlib，将 AI 技术应用于量化投资。

面向 AI 的框架 Qlib 基于现代研究工作流以模块化的方式设计，以提供最大的灵活性来适应 AI 技术。定量研究人员可以扩展模块并构建工作流，以有效地尝试他们的想法。在每个模块中，Qlib 提供了几个默认的实现选择，这些选择在实际投资中非常有效。有了这些现成的模块，定量研究人员可以专注于他们对特定模块感兴趣的问题，而不会被其他琐碎的细节分心。除了代码，计算和数据也可以在一些模块中共享，因此 Qlib 被设计成一个平台而不是一个工具箱来服务用户。

高性能基础设施 数据处理的性能对于 AI 技术等数据驱动的方法来说非常重要。作为一个面向 ai 的平台，Qlib 提供了高性能的数据基础设施。Qlib 提供了一个时间序列平面文件数据库[3]。这样的数据库专门用于金融数据的科学计算。在定量投资研究中的一些典型数据处理任务上，它的性能大大超过目前流行的通用数据库和时间序列数据库等存储解决方案。此外，该数据库提供了一个表达式引擎，可以加速因子/特征的实现和计算，这使得依赖表达式计算的研究课题成为可能。

机器学习 Qlib 的指导已经与一些典型的数据集集成，用于定量投资，典型的机器学习算法可以在其上成功学习具有泛化能力的模式。Qlib 为机器学习用户提供了一些基本的指导，并集成了一些由合理的特征空间和目标标签组成的合理任务。提供了一些典型的超参数优化工具。通过引导和合理的设置，机器学习模型可以学习具有更好泛化能力的模式，而不仅仅是过度拟合噪声。

3.2 面向 ai 的框架

图 1 展示了 Qlib 的整体框架。这个框架旨在 1)适应现代 AI 技术，2)

[3] https://en.wikipedia.org/wiki/Flat-file 数据库

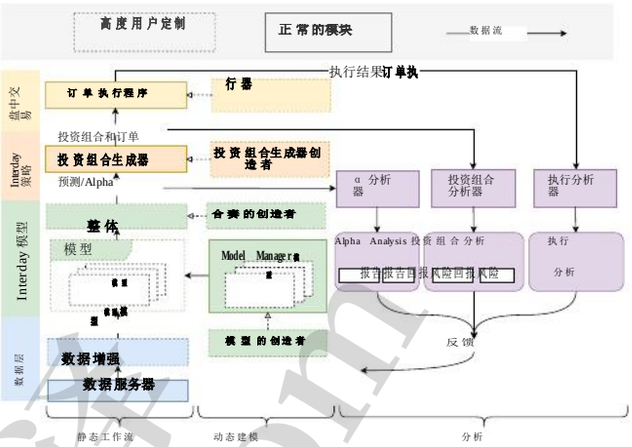

图 1:模块和用 Qlib 构建的典型工作流

帮助定量研究人员以最小的努力 3)构建一个完整的研究工作流程，并给他们最大的灵活性来探索他们感兴趣的问题，而不会被其他部分分心。

这样的目标导致了从系统设计的角度进行模块化设计。根据现代实践研究工作流程，将系统拆分为几个单独的模块。大多数量化投资研究方向，无论是传统的还是基于 ai 的，都可以看作是一个或多个模块接口的实现。Qlib 为每个模块的用户提供了几个在实际投资中表现良好的典型实现。此外，这些模块为研究人员提供了覆盖现有方法以探索新想法的灵活性。有了这样的框架，研究人员可以尝试新的想法，并以最小的成本用其他模块测试整体性能。

Qlib 的模块在图 1 中列出，并以典型的工作流程连接。每个模块对应量化投资中的一个典型子任务。模块中的一个实现可以看作是这个任务的解决方案。我们将介绍每个模块，并给出一些现有定量研究的相关示例，以展示 Qlib 如何适应它们。

首先从左下角的 **Data Server 模块**开始，它提供了一个数据引擎来查询和处理原始数据。有了检索到的数据，研究人员可以在数据**增强模块中构建自己的数据**集。研究人员已经尝试了很多解决方案，通过探索和构建有效因素/特征来构建更好的数据集[Potvin *et al.*， 2004;Neely *et al.*， 1997;Allen and Karjalainen, 1999;Kakushadze, 2016]。生成用于训练的数据集[Feng *et al.*， 2019]是提供数据集解决方案的另一个研究方向。**Model Creater 模块**基于数据集学习模型。近年来，众多研究人员探索了各种模型，从金融数据集中挖掘交易信号[Sezer *et al.*， 2019]。此外，试图学习学习的元学习[Vilalta 和 Drissi, 2002]为模型创建者模块提供了一种新的学习范式。鉴于在现代研究工作流程中有大量的方法对金融数据建模，模型管理系统已经成为 nec-

模型管理系统是工作流程中必不可少的一部分。**模型管理器模块**是为现代定量研究人员处理这类问题而设计的。在模型多样化的情况下，集成学习是增强机器学习模型性能和鲁棒性的一种相当有效的方法，在金融领域被频繁使用[Qiu *et al.*， 2014;Yang *et al.*， 2017;赵 *等*, 2017].;由 **Model Ensemble 模块**支持。**Portfolio Generator 模块**旨在从模型输出的交易信号中生成一个投资组合，这被称为投资组合管理[Qian *et al.*， 2007]。Barra [Sheikh, 1996]为这项任务提供了最流行的解决方案。对于目标投资组合，我们提供了一个高保真的交易模拟器，**Orders Executor 模块**，以检查策略的表现，而 **Analyser 模块**则自动分析交易信号、投资组合和执行结果。订单执行器模块被设计成一个响应式模拟器，而不是一个反向测试功能，它可以为一些学习范式(例如;， RL)，需要分析器模块产生的环境反馈。

量化投资数据采用时间序列格式，按时间更新。样本内数据集的大小随时间增加。利用新数据的一个典型做法是定期更新我们的模型[Wang *et al.*， 2019b]。除了更好地利用不断增加的样本内数据外，动态更新模型[Yang *等人*，2019]和交易策略[Wang *等人*，2019a]将由于股票市场的动态性质而进一步提高性能[Adam *等人*，2016]。因此，在静态**工作流** (static Workflow)中使用一套静态模型和交易策略显然不是最优解。模型和策略的动态更新是量化投资的一个重要研究方向。**动态建模**中的模块提供了适应这种解决方案的接口和基础设施。

3.3 高 性能基础设施

财 务数据

我们将在本节中总结定量研究中的数据要求。在定量研究中，最常用的数据格式遵循这样的格式

$$\text{BasicDataT}=\{x_{i,t,a}\},\quad i\in Inst, t\in \text{Time}, a\in \text{Attr}$$

其中，$x_{i,t,a}$ is 基本类型的值(例如;float, int)， Inst 表示金融工具集(例如;， 股票，期权等)，Time 表示时间戳集(例如。，时间表示股票市场的交易日，Attr 表示一种工具的可能属性集(例如。，公开价格，交易量，市场价值)，T 表示数据的最新时间戳(例如。，表示最新的交易日期。，$_{i,t,a}$ 表示工具 I 在 t 时刻的属性 a 值。

此外，工具池是指定一组随时间变化的金融工具的必要信息

$$Pool_T = \{pool_t\}, t \in Time, pool_t \subseteq Inst$$

标准普尔 500 指数 [4] 就是工具池的一个典型例子。

数据更新是一个必不可少的功能。现有的历史数据不会随着时间的推移而改变。的追加操作需要新增数据。形式化的更新操作是

$$\text{BasicDataT}=\text{OldBasicDataT}\cup\{x_{i,t,a}\}_{new}$$
$$\text{BasicDataT}_{+1}=\text{BasicDataT}\cup\{x_{i,t+1,a}\}$$
$$\text{PoolT}_{+1}=\text{PoolT}\cup\{\text{PoolT}_{+1}\}$$

用户查询可以形式化为

$$\text{DataQuery}=\{x_{i,t,a}|it\in pool_t, pool_t\in \text{Poolquery}, a\in \text{Attrquery}, timestart\leqslant t\leqslant timeend\}$$

表示特定池中特定时间范围内仪器某些属性的数据查询。

这样的要求相当简单。很多现成的开源解决方案都支持这样的操作。我们将它们分为三类，并在每个类别中列出流行的实现。

- 通 用 数 据 库 :MySQL[MySQL， 2001] ，MongoDB[Chodorow, 2013]
- 时间序列数据库:InfluxDB [Naqvi *et al.*， 2017]
- 用于科学计算的数据文件:由 numpy[Oliphant, 2006]数组或 pandas[McKinney, 2011]数据框架组织的数据

通用数据库支持不同格式和结构的数据。此外，它还提供了许多复杂的机制，如索引、事务、实体-关系模型等。它们大多给特定任务增加了沉重的依赖关系和不必要的复杂性，而不是解决特定场景中的关键问题。时间序列数据库优化了时间序列数据的数据结构和查询。但它们仍然不是为定量研究而设计的，定量研究中的数据通常是基于紧凑数组的格式，用于科学计算，以利用硬件加速。如果数据从磁盘到客户端都保持基于紧凑阵列的格式而不进行格式转换，将节省大量的时间。但是，无论是通用型数据库还是时间序列型数据库，都是以不同的格式存储和传输数据的通用型，这对于科学计算来说效率很低。

由于数据库的低效率，基于阵列的数据在科学界得到了普及。Numpy 数组和 pandas 数据帧是科学计算中的主流实现，通常以 HDFor[5] 存储在磁盘 [6] 上。这样格式的数据依赖性很轻，对于科学计算来说非常高效。但是，这样的数据存储在单个文件中，很难更新或查询。

经过对上述存储方案的考察，我们发现没有一个能够很好地适应定量研究场景。有必要为定量研究设计一个定制化的解决方案。

文 件存储设计

图 2 演示了文件存储设计。如图左侧所示，Qlib 以树形结构组织文件。数据根据不同被分为文件夹和文件

[4] https://en.wikipedia.org/wiki/S%26P 500 指数

[5]

[6] https://en.wikipedia.org/wiki/Hierarchical 数据格式
https://docs.python.org/3/library/pickle.html

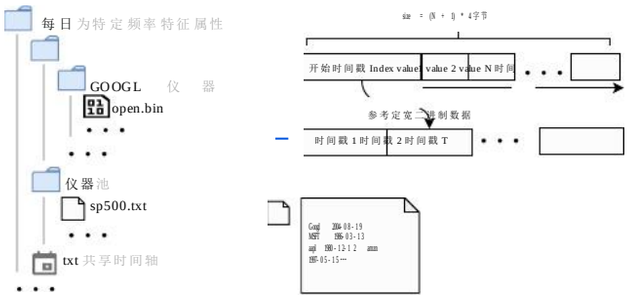

图 2:平面文件数据库的描述;左边部分是文件的结构;右边部分是文件的内容

频率、仪器和属性。所有属性的值都以紧凑的固定宽度格式存储在二进制数据中，因此按字节进行索引成为可能。共享时间轴被单独存储在一个名为“calendar.txt”的文件中。属性值的数据文件将其前 4 个字节设置为时间轴的索引值，以指示该系列数据的开始时间戳。有了开始时间索引，Qlib 可以在时间维度上对齐所有的值。

数据以紧凑的格式存储，可以高效地组合成数组进行科学计算。在科学计算中实现了类似数组数据的高性能的同时，也满足了量化投资场景下的数据更新需求。所有数据都按照时间顺序排列。新的数据可以通过追加来更新，效率相当高。添加和删除属性或工具是相当直接和高效的，因为它们存储在单独的文件中。这样的设计非常轻量。没有数据库的开销，Qlib 实现了高性能。

表达式引擎

基于基本数据开发新的因子/特征是一项相当常见的任务。这样的任务占用了许多定量研究人员的很大一部分时间。都通过代码实现这样的因子，计算过程很耗时。因此，Qlib 提供了一个表达式引擎来最大限度地减少这类任务的工作量。

实际上，因子/特征的本质是将基本数据转换为目标值的函数。这个函数可以分解成一系列表达式的组合。表达式引擎就是基于这个思想设计的。有了这个表达式引擎，定量研究人员可以通过编写表达式而不是复杂的代码来实现新的因素/特征。例如，布林带技术指标[Bollinger, 2002]是一种广泛使用的技术因子，它的上界可以用表达式引擎实现，只需一个简单的表达式 *“(MEAN($close, N)+2*STD($close, N)-$close)/MEAN($close, N)”*。

这样的实现简单、易读、可重用、可维护。用户只需一系列简单的表达式就可以轻松构建一个数据集。通过搜索表达式来构建有效的交易信号是一个典型的研究课题，许多研究者对此进行了探索[Allen and Karjalainen, 1999;Neely *et al.*, 1997;Potvin *et al.*, 2004]。一个表达式

对于这样一个研究课题，引擎是一个必不可少的工具。

缓存系统

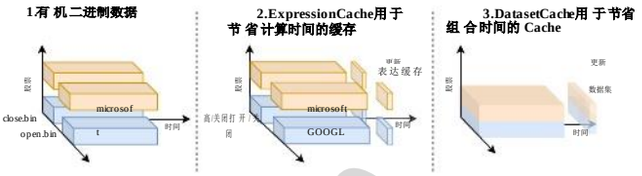

图 3:Qlib 的磁盘缓存系统;用于节省表达式计算时间的表达式缓存;数据集缓存，用于节省数据组合的时间

为了避免重复计算，Qlib 有一个内置的缓存系统。它由内存缓存和磁盘缓存组成。

内存缓存当 Qlib 用它的表达式引擎计算因子/特征时，它将表达式解析为语法树。节点的所有计算结果将存储在内存中的 LRU(Least Recently Used，最近最少使用)缓存中。相同(子)表达式的复制计算可以被保存。

在量化投资中，数据处理的一个典型工作流可以分为三个步骤:获取原始数据，计算表达式，并将数据组合成数组进行科学计算。计算表达式和组合数据是非常耗时的。如果我们可以缓存共享的中间数据，可以节省很多时间。在实际的数据处理任务中，很多中间结果是可以共享的。例如，相同的表达式计算可以被不同的数据处理任务共享。因此 Qlib 设计了 2 级磁盘缓存机制。缓存系统如图 3 所示。左边的部分是我们在 3.3 节中描述的原始数据。第一级是表达式缓存，它会将所有计算出来的表达式保存到磁盘缓存中。表达式缓存的数据结构与原始数据相同。使用表达式缓存，相同的表达式只会计算一次。表达式缓存之后是数据集缓存，它存储合并后的数据，以节省合并时间。两层的缓存数据都是按时间排列的，在时间维度上是可索引的，所以即使查询时间发生变化，也可以共享磁盘缓存。而且，由于数据按时间排列，Qlib 支持通过追加新数据的方式进行数据更新。有了这样的机制，数据的维护就容易多了。

3.4 Guidance for Machine Learning

正如我们在第 2 节中讨论的，对机器学习算法的指导是非常重要的。Qlib 为机器学习算法提供了典型的数据集。在 Qlib 中可以找到一些典型的任务设置，比如数据预处理、学习目标等。研究人员不需要从头开始探索所有问题。这样的指南为研究人员在这个研究领域开始他们的旅程提供了大量的领域知识。

对于大多数机器学习算法来说，超参数优化是实现更好泛化的必要步骤。虽然它很重要，但它需要付出很多努力，而且确实如此

	HDF5	MySQL	MongoDB	InfluxDB	Qlib -E -D	Qlib +E -D	Qlib +E +D
存储(MB)	**287**	1332 年	911	**394**	**303**	802	1000 年
加载数据(s)	**0.80** ± **0.22**	182.5 ± 4.2	70.3 ± 4.9	186.5 ± 1.5	**0.95** ± **0.05**	4.9 ± 0.07	7.4 ± 0.3
计算 Expr。(s)	179.8 ± 4.4				137.7 ± 7.6	**35.3** ± **2.3**	-
将指数(s)	-				3.6 ± 0.1		-
过滤池(s)	3.39±0.24						-
结合数据(年代)	1.19 ± 0.30						-
总数(1CPU) (s)	184.4 ± 3.7	365.3 ± 7.5	253.6 ± 6.7	368.2 ± 3.6	147.0 ± 8.8	47.6 ± 1.0	**7.4** ± **0.3**
总 cpu (64) (s)	-				8.8 ± 0.6	**4.2** ± **0.2**	-

表 1:不同存储方案性能比较

完全重复。因此，Qlib 提供了一个**超参数调优引擎(Hyperparameters Tuning Engine, HTE)**来简化这样的任务。HTE 提供了一个接口来定义一个超参数搜索空间 Θ，然后自动搜索最佳超参数 Θ。

在一个典型的建模时间序列数据的金融任务中，新的数据是按时间顺序来的。为了利用新数据，必须定期对模型进行新数据的重新训练。新的最佳超参数 θ 会发生变化，但往往接近于之前的最佳超参数。HTE 提供了一种专门用于金融任务超参数优化的机制。它为超参数搜索空间生成了一个新的分布，以更好的机会以更少的试验达到最佳点。搜索 θ 的分布可以形式化为

$$p_{new}(x) = \frac{p_{prior}(x)\varphi_{\theta_{prev},\sigma^2}(x)}{\mathbb{E}_{x\sim p_{prior}}[\varphi_{\theta_{prev},\sigma^2}(x)]}$$

其中，prior pi 是原始超参数搜索空间;ϕ θprev，σ2(x)~ N(θ prev， σ 2); θ prev是上次模型训练中最好的超参数。超参数搜索空间的域保持不变，但 θ prev 周围的概率密度增大。

4 用例与性能评估 4.1用例

Qlib 提供了一个**配置驱动的管道引擎(CDPE)**来帮助研究人员更容易地构建图 1 所示的整个研究工作流程。用户可以用一个简单的配置文件来定义一个工作流，比如 List **??**(一些琐碎的细节被“…”取代)。这样的界面并不是强制性的，我们把最大的灵活性留给用户，让他们像积木一样通过代码构建定量研究工作流。

4.2 性能评估

数据处理的性能对于 AI 技术等数据驱动的方法非常重要。Qlib 作为一个面向 ai 的平台，为数据存储和数据处理提供了解决方案。为了演示 Qlib 的性能，我们将 Qlib 与 3.3 节中讨论的其他几个解决方案进行了比较，其中包括 *HDF5*、*MySQL*、*MongoDB*、*InfluxDb 和 Qlib*。*Qlib +E -D* 表示启用表达式缓存和禁用数据集缓存的 Qlib，以此类推。

```
model:
  class:   "qlib.model.GBDTModel"
  args:    ...
data:
  class:   "qlib.dataset.Alpha360"
learner:
  class:   "qlib.trainer.NormalTrainer"
  args:    ...
portfolio_manager:
  class:   "qlib.portfolio.TopkStrategy"
executor:
  account: ...
```

图 4:CDPE 的一个配置示例

解决方案的任务是从股票市场的基本 OHLCV[7] daily 数据中创建数据集，涉及到数据查询和处理。最终的数据集由来自 OHLCV 数据的 14 个因子/特征组成(例如:“性病(近美元,5)/ $关闭”)。数据的时间范围为 1/1/2007 - 1/1/2020。股票池每天由 800 只股票组成，每天都在变化。

除了比较每个解决方案的总时间外，我们还将任务分解为以下步骤来了解更多细节。

- **Load Data** 将 OHCLV 数据或缓存加载到 RAM 中，作为基于数组的格式进行科学计算。
- **计算 Expr。**计算导出的因子/特征。
- **转换索引** 它只适用于 Qlib。因为 Qlib 不存储索引(即。，时间戳，股票 id)在原始数据中，它必须设置数据索引。
- **过滤数据** 按特定池对股票数据进行过滤。例如，SP500 总共涉及 1000 只以上的股票，但它每天只包含 500 只股票。具体某一天没有纳入 SP500 的数据应该被过滤掉，尽管它曾经纳入过 SP500。在加载数据时不可能过滤掉数据，因为一些派生的特征依赖于 OHLCV 的历史数据。
- **组合数据将**不同股票的所有数据串联成一个单一的基于数组的数据

正如我们在表 1 中所看到的。Qlib 的紧凑存储实现了与专用科学文件相似的大小和加载速度

[7] 股票的开盘价、最高价、最低价、收盘价和交易量

HDF5 数据文件。数据库加载数据的时间太长。在研究了底层实现后，我们发现无论是通用数据库还是时间序列数据库解决方案，数据都要经过太多的接口层和不必要的格式转换。这样的开销大大降低了数据加载过程的速度。由于 Qlib 的内存缓存，Qlib -E -D 节省了 Compute Expr 大约 24% 的时间。而且，Qlib 提供了表达式缓存和数据集缓存机制。在 Qlib +E -D 中启用表达式缓存后，80.4%的时间用于 Compute Expr。如果没有错过表达式缓存，则保存。将因子/特征组合为每个股票的一个基于数组的数据占了 Qlib +E -d 的主要时间消耗，它包含在 Compute Expr 中。的一步。除了计算成本，最耗时的步骤是数据组合。数据集缓存的设计就是为了减少这样的开销。如 Qlib +E +D 列所示，时间成本进一步降低。

而且，Qlib 可以利用多个 CPU 核心来加速计算。正如我们在表 1 的最后一行中所看到的，对于具有多个 cpu 的 Qlib 来说，时间成本显著降低。Qlib +E +D 不能进一步加速，因为它只是读取现有的缓存，几乎什么都不计算。

4.3 更多关于 Qlib 的知识

Qlib 是一个持续发展中的开源平台。更详细的文档可以在其 github 存储库 [8] 中找到。很多功能(例如;本文没有详细介绍的数据服务与客户-服务器架构、分析系统、云上自动部署)在在线知识库中都可以找到。欢迎赐稿。

5 的结论

在本文中，我们提出了 AI 时代现代定量研究者的实际问题。基于这些实际问题，我们设计并实现了 Qlib，旨在让每一个量化研究人员都能意识到 ai 技术在量化投资中的巨大潜力。

参考文献

[亚当 *等人*，2016]克劳斯·亚当，阿尔伯特·马塞，胡安·巴勃罗·尼科里尼。*股市波动与学习*，2016 年。

[Allen and Karjalainen, 1999] Franklin Allen and Risto Karjalainen。利用遗传算法寻找技术交易规则。*金融经济学报*，51(2):245-271,1999。

[博林格，2002]约翰·博林格。*布林杰论布林杰带*。McGraw Hill Professional，2002 年。

[Chodorow, 2013] Kristina Chodorow。*MongoDB:权威指南:强大且可扩展的数据存储。*”O 'Reilly Media, Inc.”，2013 年。

[邓 *等*，2016]邓悦，鲍峰，孔友勇，任志泉，戴琼海。用于金融信号表示和交易的深度直接强化学习。*神经网络与学习系统 IEEE 汇刊*，28(3):653-664,2016。

[Feng *et al.*，2019]冯富力，陈慧敏，何向南，丁吉，孙茂松，蔡达生。用对抗性训练增强股票运动预测。*第 28 届国际人工智能联合会议论文集*，第 5843-5849 页。AAAI 出版社，2019 年。

[弗斯，2004]N 弗斯。为什么要使用 quantlib。*纸可用在:http://www。quantlib。英国公司/出版/ quantlib。pdf*,2004 年。

[卡库沙泽，2016]祖拉·卡库沙泽。101 公式化 alpha。*威尔莫特*，2016(84):72-81,2016。

[Li *等人*，2016]李斌，Doyen Sahoo, Steven CH Hoi。Olps:一个在线投资组合选择的工具箱。*《机器学习研究杂志》*，17(1):1242-1246,2016。

[McKinney, 2011]韦斯·麦金尼(Wes McKinney)。Pandas:用于数据分析和统计的基础 python 库。*Python for High Performance and Scientific Computing*，2011 年 14 日。

[Murphy, 1999] John J Murphy。*金融市场的技术分析:交易方法和应用的综合指南*。企鹅出版社,1999 年。

[MySQL, 2001] AB MySQL。Mysql，2001 年。

[Naqvi *等人*，2017]Syeda Noor Zehra Naqvi, Sofia Yfanti-dou，和 Esteban Zim´anyi。时间序列数据库和 in-fluxdb。*Studienarbeit，布鲁塞尔自由大学*，2017。

[Neely *et al.*，1997] Christopher Neely, Paul Weller, and Rob Dittmar。外汇市场的技术分析有利可图吗?一种基因编程的方法。*《金融与定量分析》*，32(4):405-426,1997。

[奥列芬特，2006]Travis E Oliphant。*NumPy 指南*，第 1 卷。美国 Trelgol 出版公司，2006 年。

[佩特科娃，2006]雷丽莎·佩特科娃。fama-french 因子能代表预测变量的创新吗?《金融 *学报*》，61(2):581-612,2006。

[Potvin *et al.*，2004] Jean-Yves Potvin, Patrick Soriano, and Maxime Vall´ee。用基因编程在股票市场上生成交易规则。*计算机与运筹学*，31(7):1033-1047,2004。

[Qian *et al.*，2007] Edward E Qian, Ronald H Hua，和 Eric H Sorensen。*量化股票投资组合管理:现代技术与应用*。CRC 出版社，2007 年。

[邱 *等*，2014]邱学恒，张乐，任叶，Pon-nuthurai N Suganthan, Gehan Amaratunga。用于回归和时间序列预测的集成深度学习。*2014 年 IEEE 集成学习中的计算智能研讨会(CIEL)*，第 1-6 页。IEEE 2014。

[8] https://github.com/microsoft/qlib/

[Sezer *等人*，2019]Omer Berat Sezer, Mehmet Ugur Gudelek, Ahmet Murat Ozbayoglu。深度学习的金融时间序列预测:系统文献综述:2005-2019。*arXiv 预印本 arXiv:1911.13288*，2019。

[谢赫，1996]阿米尔·谢赫。巴拉的风险模型。*《巴拉研究洞察》*，1996年第1-24页。

[维拉塔和德里西，2002]里卡多·维拉塔和尤瑟夫·德里西。元学习的透视与调查。*人工智能评论*，18(2):77-95,2002。

[Wang *et al.*， 2019a]王乐文，刘维青，杨晓，边江。保守还是激进?有信心的动态投资组合构建。*2019 年 IEEE 信号与信息处理全球会议(GlobalSIP)*，第 1-5 页。IEEE 2019。

[Wang *et al.*， 2019b]王守祥，王璇，王少民，王丹。基于注意机制和滚动更新的双向长短期记忆方法，用于短期负荷预测。*国际电力与能源系统学报*，109:470-479,2019。

[杨 *等*，2017]杨兵，龚子佳，杨文琪。使用深度神经网络集合的股市指数预测。*2017 年第 36 届中国控制大会(CCC)*， 3882-3887 页。IEEE 2017。

[Yang *et al.*， 2019]杨晓，刘维青，王乐文，曲程，边江。基于注意力的多重投资策略组合的分治框架。*2019 年 IEEE 信号与信息处理全球会议(GlobalSIP)*，第 1-5 页。IEEE 2019。

[Zhao *et al.*， 2017]赵杨，李建平，余精益。原油价格预测的深度学习集成方法。*能源经济*，66:9-16,2017。