数据底座洞察 03

Jiangsheng Yu

03/05/2024

三个主要问题

1) 数据的转化率：数据质量的评判标准是什么？如何控制数据质量？面对数据荒，如何解决这个问题？
2) 迁移学习：中小型公司缺少数据，如何让它们用起LLM形成闭环？
3) 分布式计算：Anyscale 公司的核心技术？MosaicML如何大幅度降低机器学习的成本？

数据质量的评判标准

1. **准确性（Accuracy）**：数据与真实情况的一致程度。准确的数据能够反映实际情况，有助于做出正确的决策。
2. **完整性（Completeness）**：数据集中是否包含了所有相关信息，是否缺少重要的数据记录或字段。完整的数据能够提供全面的信息支持。
3. **一致性（Consistency）**：数据在不同数据源、不同时间点或不同系统中的一致性。数据应该保持一致，避免出现矛盾或冲突。
4. **及时性（Timeliness）**：数据的更新频率和及时程度。及时的数据能够反映当前的情况，对实时决策和监控至关重要。
5. **唯一性（Uniqueness）**：数据集中是否存在重复记录，以及如何处理这些重复记录。确保数据唯一性可以避免数据冗余和混淆。
6. **合法性（Legitimacy）**：数据的来源合法、采集过程合规，并符合相关法律法规和策略要求。合法的数据能够保证数据的可信度。
7. **有效性（Validity）**：数据是否符合预期的使用目的，并能够提供有用的信息。有效的数据能够支持业务需求和决策。

如何控制数据的质量？

1. **数据标准化**：确保数据按照一致的标准进行录入和存储，包括数据格式、单位、命名规范等。这有助于提高数据的可比性和可理解性。
2. **数据采集和验证**：为数据处理人员提供培训，使他们了解数据质量的重要性，并掌握相应的数据处理技能和最佳实践。在数据采集和录入阶段，实施验证措施以确保数据的准确性。这包括验证数据格式、范围、唯一性等。建立数据审查和验证流程，确保数据在不同阶段经过审查和验证。
3. **数据清洗**：清洗数据是指识别和纠正数据中的错误、缺失、重复或不一致的部分。使用数据清洗工具和算法可以自动化这一过程。
4. **建立数据质量度量指标**：制定适当的数据质量度量指标，以评估数据的准确性、完整性、一致性、及时性等方面。这些指标可以用于监控数据质量，并识别潜在的问题。
5. **建立数据质量管理团队**：建立专门的团队或角色负责数据质量管理，监督数据质量相关的工作，并及时处理数据质量问题。定期评估和审查数据质量控制措施的有效性，并根据反馈和经验不断改进和优化这些措施。

如何解决数据荒？

1. **数据采集**：扩大数据获取渠道，包括积极收集内部数据、利用外部数据源、合作伙伴数据共享等方式获取更多的（垂域）数据。可以考虑利用网络爬虫、API接口、数据交换协议等技术手段获取外部数据。
2. **数据填充**：利用数据合成技术，通过模型估算或者插值方法填充缺失的数据，以提高数据的完整性。这包括利用机器学习模型、时间序列模型等方法来估计出**缺失数据**。
3. **数据生成**：利用**生成式AI**的新技术来解决数据荒问题，如GAN、VAE、diffusion model、flow-based methods等。

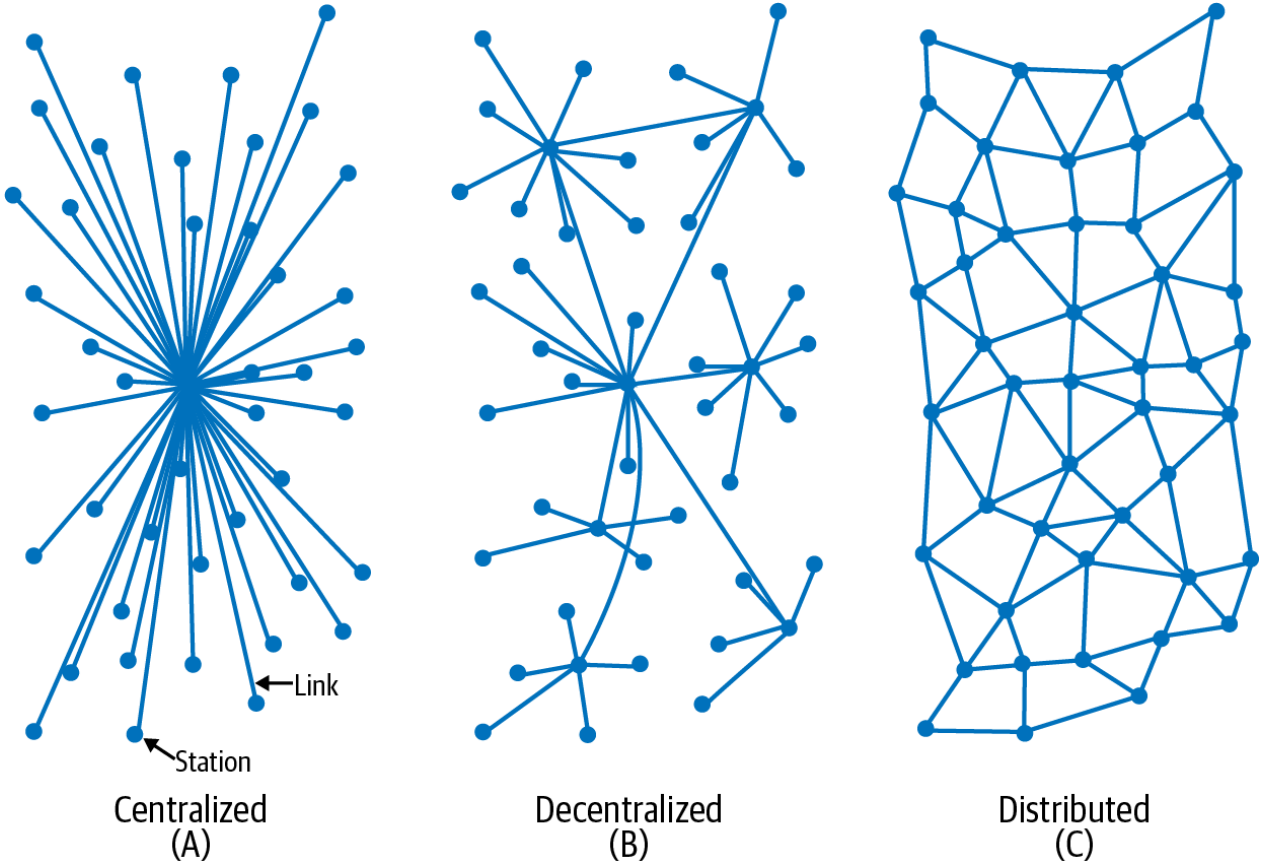
Anyscale 公司的核心技术：分布式计算

Anyscale公司专注于为开发人员提供用于构建、部署和管理分布式应用程序的工具和平台。其中，Anyscale 最著名的核心技术之一是 Ray。

- Ray 是一个高性能分布式执行框架，旨在简化分布式应用程序的开发过程，并提供可扩展的、高效的分布式计算能力。
- Ray 提供了一个简单而灵活的 API，可以用于构建各种类型的分布式应用程序，包括机器学习训练、流式数据处理等

下一跳：**去中心化的分布式算力众筹+区块链+可信计算**。让计算资源被充分利用，实现共赢。即所谓“我为人人，人人为我”。

去中心化的算力众筹



- 分布式计算的摩尔定律：每18个月，“算力”提升10倍。
- 算力不应该被集中到少数大厂，而应该变得更加亲民才有可能让AI得以蓬勃发展。

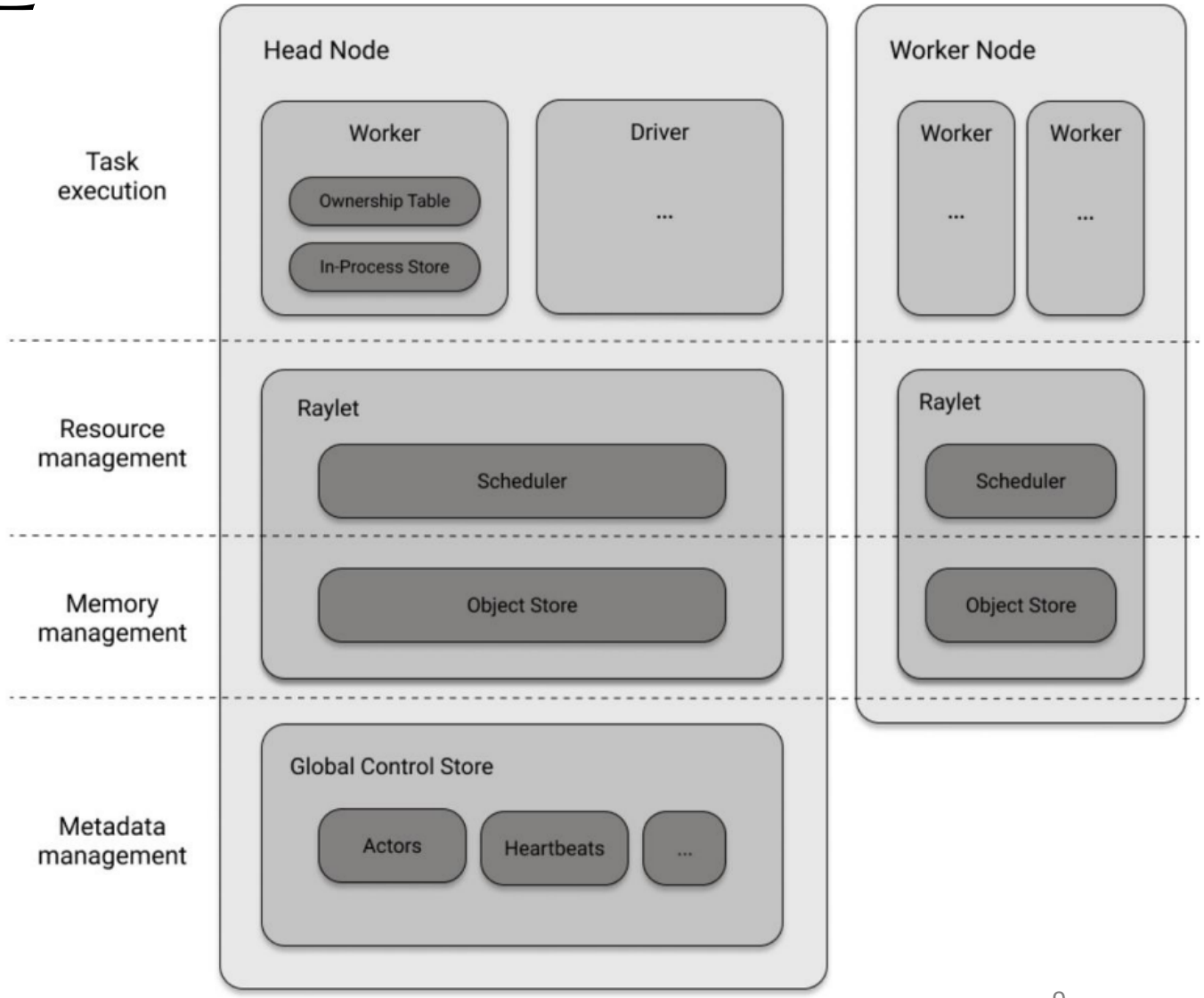
1) 每个节点都可以发起计算任务，在计算网络中通过“资源账本”将任务分发下去。
2) 计算网络是动态的，节点将闲置算力释放给网络，也可以从网络中得到“奖励”。
3) 借助网络通讯的进步，让算力在网络中流动变得更加智能化、自动化，提升大模型的训练效果，同时降低训练成本。
4) 算力众筹的平台可以形成共赢的生态。优化公司内部的算力成本。

Ray的功能和特性

1. **分布式任务调度和执行**：Ray 提供了分布式任务调度和执行的能力，可以自动将任务分配到可用的计算资源，并管理任务的执行过程。
2. **弹性扩展和容错处理**：Ray 支持弹性扩展和容错处理，可以根据需求动态扩展计算资源，并在节点故障时自动进行容错处理。
3. **分布式状态管理**：Ray 提供了分布式状态管理的能力，允许用户在分布式任务之间共享和管理状态信息。
4. **高性能和低延迟**：Ray 针对分布式计算场景进行了优化，具有高性能和低延迟的特点，能够在大规模数据和计算量下保持高效率。
5. **生态系统支持**：Ray 生态系统提供了丰富的工具和库，支持机器学习（包括强化学习、深度学习）、流式数据处理等多种应用场景。

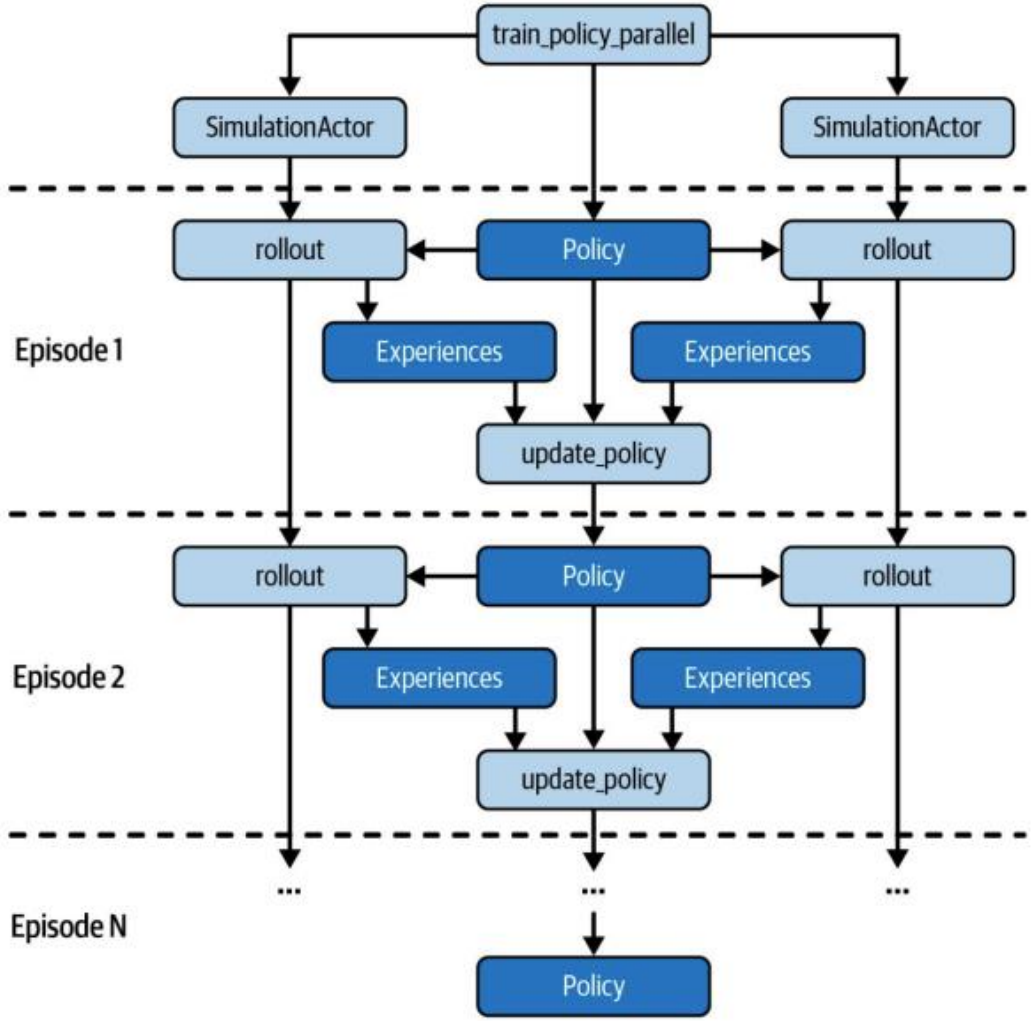
Ray 的架构组件概述

- 每个 Ray 集群都有一个特殊的节点，称为头节点。
- 头节点还带有一个称为全局控制存储（GCS）的组件。 GCS 是目前在 Redis 中实现的键 值存储。它是一个重要的组件，承载有关集群的全局信息，例如系统级元数据。例如，它有一个表， 其中包含每个 Raylet 的心跳信号，以确保它们仍然可达。
- Raylet 反过来向 GCS 发送心跳信号 以表明它们还活着。 GCS还将Ray actor和大对象的位置存储在各自的表中，并了解对象之间的依赖关系。



Ray 并行训练强化学习策略

- train_policy_parallel函数创建多个SimulationActor，以及带有create_policy 的策略。将策略放入对象存储中。
- 模拟参与者根据策略创建部署，从而收集经验，update_policy用于更新策略。
- 由于更新策略的设计方式，这种方法有效。如果经验是通过一次或多次模拟收集的也没问题。
- 持续推出和更新，直到满足条件，然后返回最终的trained_policy。可以从对象存储中检索已完成的部署并用于更新策略。
- 本质上，并行化训练过程所需要做的就是以合适的方式在类上使用ray.remote，然后使用正确的远程调用。

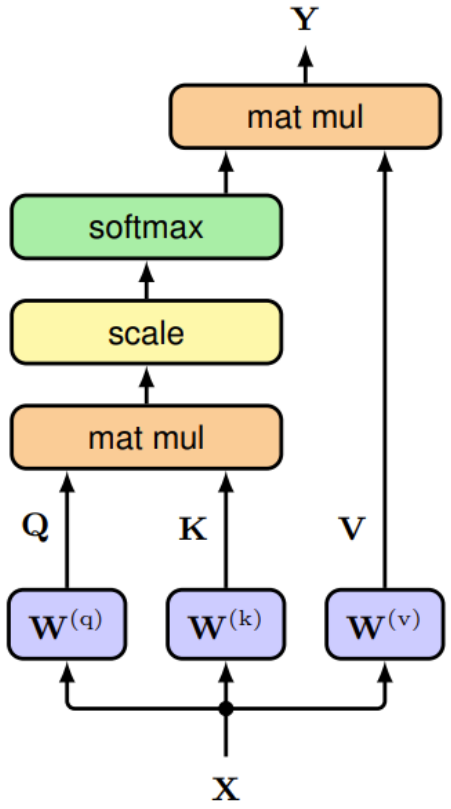


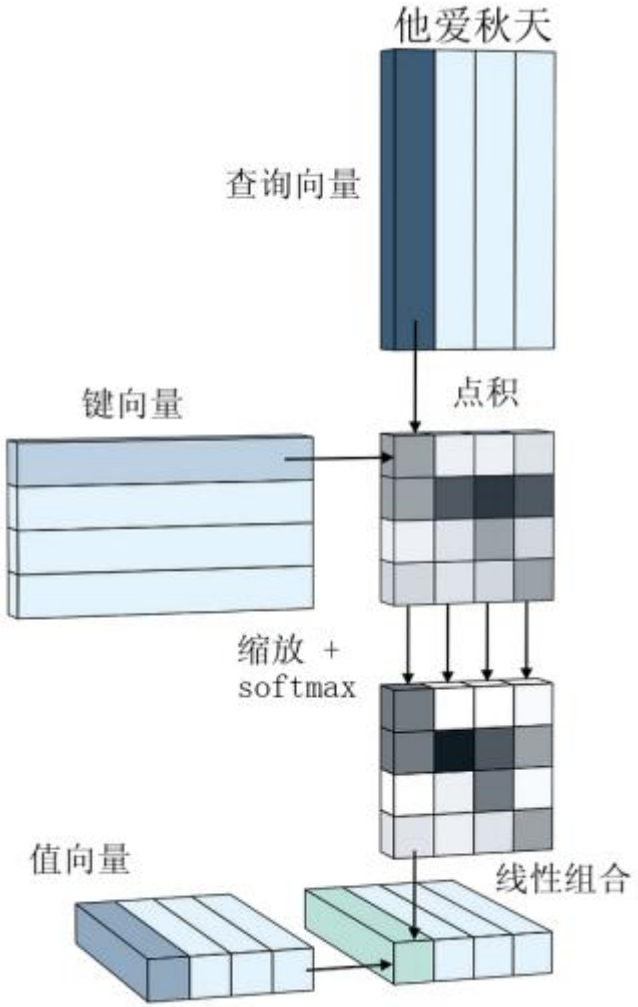
Ray与Spark的对比

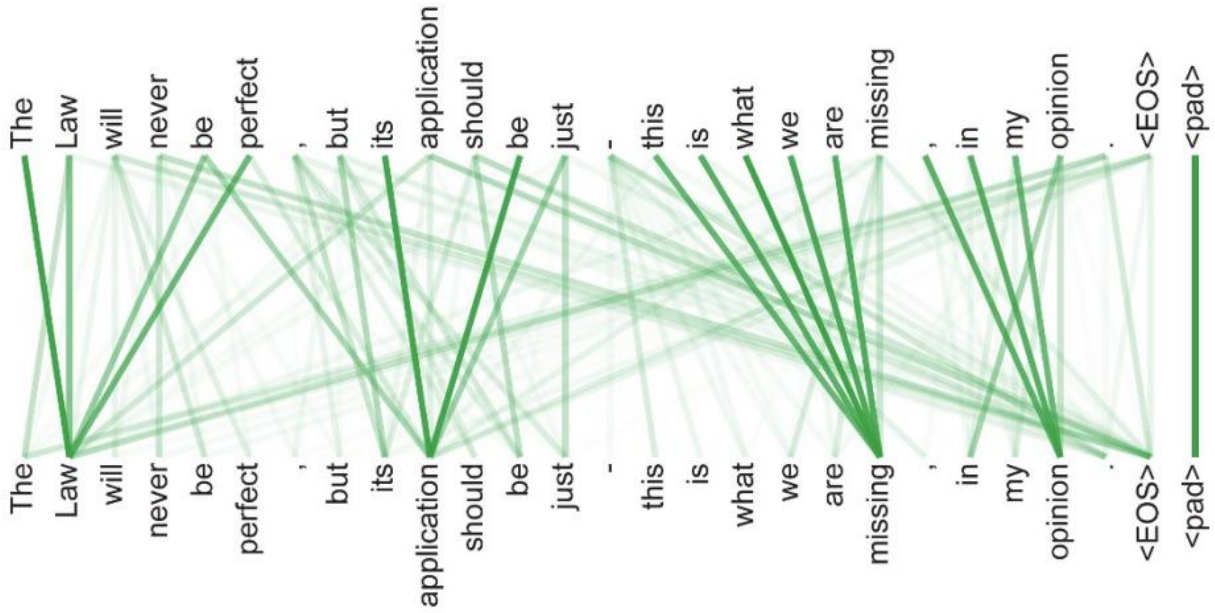
- General-purpose distributed Framework
  - Tasks & Actors as low-level primitives: Fine-grained APIs
  - Framework to build other distributed frameworks
- Distributed Scheduler
  - DAGs are dynamic
- Distribute Object Store
- Not to displace any libraries
  - Interoperate & Integrate
- Asynchronous framework
- Native ML library integrations
  - Scale ML/DL/RL workloads
- Pythonic in all its aspects

- Powerful & performant specific-purpose distributed Framework
  - DataFrame Abstraction
  - Coarse grained API (DSL)
- Static Scheduler
  - DAGs statically computed
  - Lazy evaluation
- Best-of-the-breed for Data Analytics at Scale
  - ETL, ELT, SQL, Streaming,
  - Building Lakehouse
- Synchronous framework
- Supports DL and integration w/
  - Horovod, PyTorch & TensorFlow
- PySpark

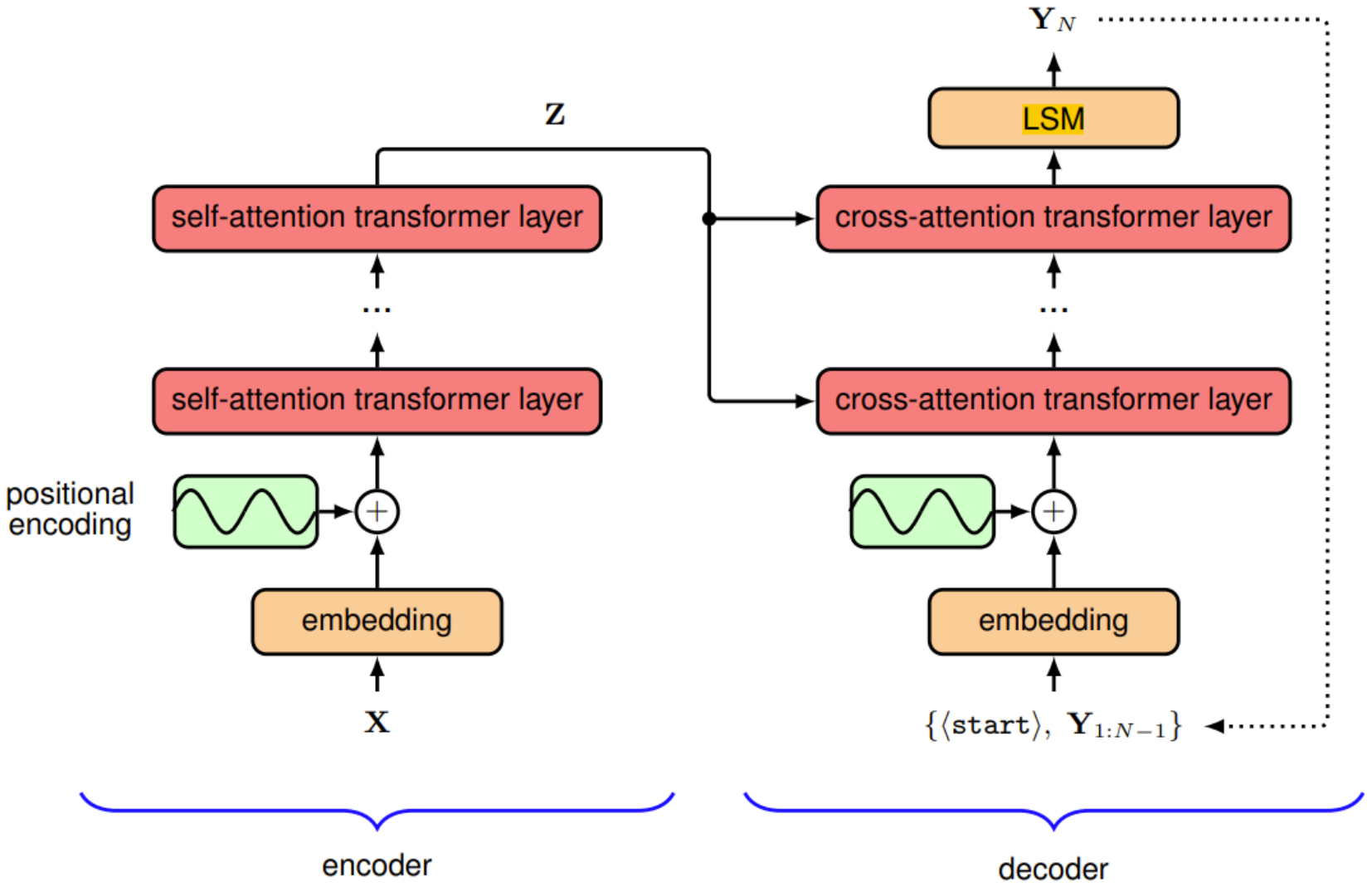
Transformer：引入变换，深挖关联性







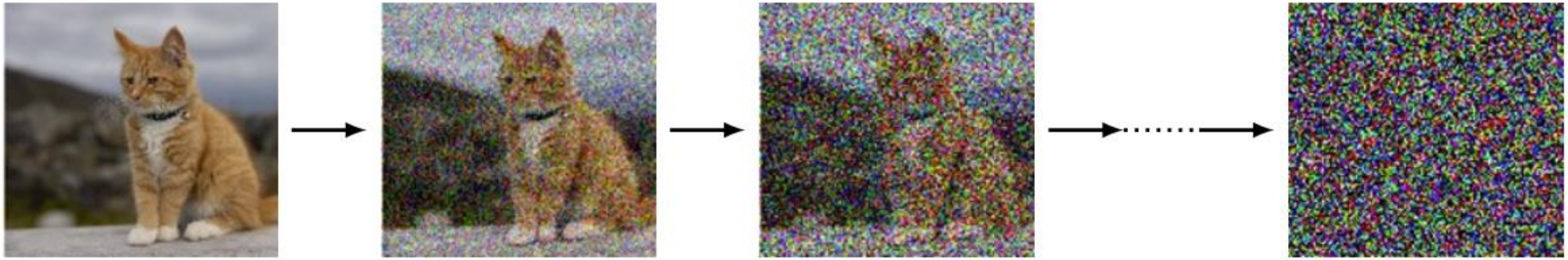
Transformer的架构

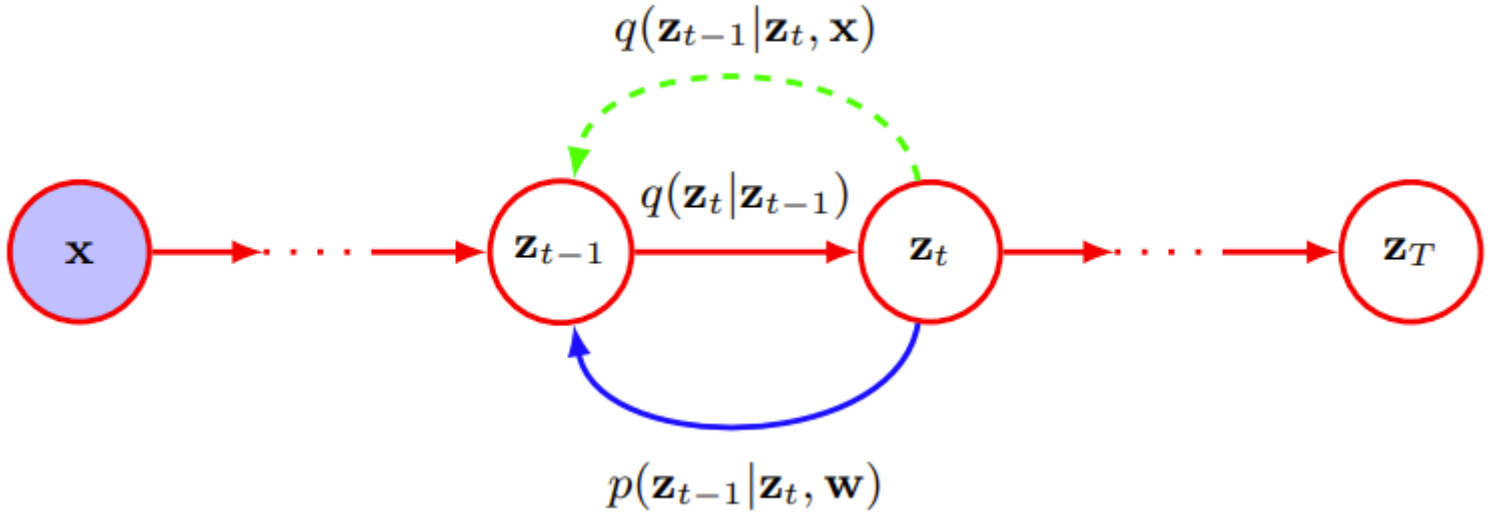


目标：序列的生成
手段：自回归模型

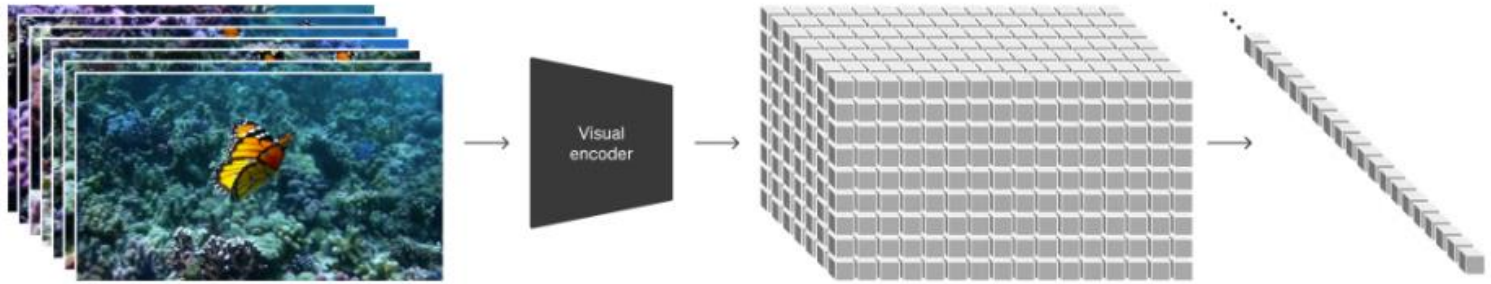
下一跳：引入隐状态；词汇语义向量；更高效的长记忆；知识的向量表示等。
理由：隐状态利于知识、推理、揭示因果关系等。

扩散模型：生成图片


$\mathbf{x}$
$\mathbf{z}_1$
$\mathbf{z}_2$
$\mathbf{z}_T$


$q(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{x})$
$q(\mathbf{z}_t|\mathbf{z}_{t-1})$
$\mathbf{x}$
$\mathbf{z}_{t-1}$
$\mathbf{z}_t$
$\mathbf{z}_T$
$p(\mathbf{z}_{t-1}|\mathbf{z}_t, \mathbf{w})$

Sora=Diffusion+Transformer：还是关联性！



作为模拟器，Sora 目前表现出许多局限性。例如，它不能准确地模拟许多基本相互作用的物理过程，例如玻璃破碎。其他交互（例如吃食物）并不总是会产生对象状态的正确变化。视频生成面临的一个重大挑战是在采样长视频时保持时间一致性。

对Sora能否充当世界模拟器，学术界还有争议。见Yann LeCun的观点。

Sora将带来视频革命

- OpenAI的Sora将改变视频产业。Youtube、字节、Meta等拥有视频大数据的公司将产生激烈竞争。
- 自动驾驶将因为视频理解能力的提升而跃迁至L5。
- 深伪技术将变得更加泛滥而难以控制。
- 游戏、电影、视频加工等产业将重新定义。
- 机器数据将发生大爆炸。将会产生一些专门生成数据的新兴产业。
- 算力诉求变得更加急迫。
- 高质量的数据和关键算法成为难得一求的商品。
- 轻量模型的挑战日趋尖锐。

大语言模型存在的问题

- 尽管LLM 的性能令人印象深刻，但它们仍然存在一些问题，例如：它们经常自信地给出问题的错误答案（即“幻觉”）。它们可以产生有偏差的输出。
- LLM 背后的基本思想非常简单（即自回归transformer的最大似然训练），并且可以用大约 300 行代码来实现。仅通过扩大模型和训练数据，似乎就会有质的飞跃。
- 然而，我们尚不清楚这种学习能力足够撑起AGI，机器需要从经验中衍生出对世界更深入的、非语言的理解。

降低ML训练和推理的成本

- 在基础模型上，利用“专有数据”进行微调。这些专有数据可以生成。
- 矩阵的符号计算和微分计算。
- 用变分法解决最优化问题（例如最速降线，在函数空间里搜索最优解）。参考 M. Jordan团队的工作。
- 深度神经网络的蒸馏。
- 蒙特卡罗方法（我在写一本书）。
- 联邦学习。
- 大模型的趋势：引入更多更合理的潜在变量增强对关联性（甚至因果性）的描述能力。

Input: Training data $\mathcal{D} = \{\mathbf{x}_n\}$
Noise schedule $\{\beta_1, \ldots, \beta_T\}$
Output: Network parameters $\mathbf{w}$

---

for $t \in \{1, \ldots, T\}$ **do**
  $\alpha_t \leftarrow \prod_{\tau=1}^{t}(1 - \beta_\tau)$ // Calculate alphas from betas
end for
repeat
  $\mathbf{x} \sim \mathcal{D}$ // Sample a data point
  $t \sim \{1, \ldots, T\}$ // Sample a point along the Markov chain
  $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{\epsilon}|\mathbf{0}, \mathbf{I})$ // Sample a noise vector
  $\mathbf{z}_t \leftarrow \sqrt{\alpha_t}\mathbf{x} + \sqrt{1 - \alpha_t}\boldsymbol{\epsilon}$ // Evaluate noisy latent variable
  $\mathcal{L}(\mathbf{w}) \leftarrow \|\mathbf{g}(\mathbf{z}_t, \mathbf{w}, t) - \boldsymbol{\epsilon}\|^2$ // Compute loss term
  Take optimizer step
until converged
return $\mathbf{w}$

Training a denoising diffusion probabilistic model (DDPM)

有关关联性的理论研究

- 大矩阵计算中的基本算子的优化与实现。
- 大算力之下的数值分析和变分优化理论。
- 随机模拟技术：在近似计算中的应用、构建世界模型等。
- 分层规划：如Yann LeCun的JEPA (Joint-Embedding Predictive Architecture)。
- 元学习（meta learning）方法。
- 过程类、算法类数据的收集与研究。
- 从观测到知识表示：因果推断。
- 生成式AI的统计方法、连接主义方法等。
- 高维数据分析，包括特征工程。
- 基于生成式AI的自动推理技术。