HiPPO: Recurrent Memory with Optimal Polynomial Projections
HiPPO: 具有最优多项式投影的循环记忆

Albert Gu*, Tri Dao*, Stefano Ermon, Atri Rudra, and Chris Ré

机器学习的诸多领域都需要以在线的方式处理序列数据。例如，在实时观察时间序列的同时需要持续预测未来，在部分可观测环境中的智能体必须学会将其累积经验编码为状态，以便于导航和决策。在建模长期和复杂的时间依赖性时，最根本的问题在于记忆：如何存储并整合先前时间步的信息。然而，流行的机器学习模型往往存在*遗忘问题*：它们要么使用固定大小的上下文窗口（如注意力机制），要么采用经验上存在有限记忆范围的启发式机制（由于"梯度消失"问题）。

本文介绍了我们从基本原理出发，解决增量维护序列记忆表示这一根本问题的方法：

1. 找到一个**可以数学分析的技术表述**，并通过 HiPPO 框架推导出闭式解。

2. 展示我们的方法如何轻松集成到端到端模型（如 RNN）中，我们的框架既概括了之前的模型（包括流行的 LSTM 和 GRU），又对它们进行了改进，在衡量长程记忆的流行基准测试排列的 MNIST 上达到了最先进的水平。

3. 展示框架洞见如何揭示具有独特理论性质的方法——我们重点介绍了一个特定模型**HiPPO-LegS**，它计算效率高，可证明地缓解了梯度消失问题，并且是首个已知的展现"时间尺度鲁棒性"的方法！

我们的论文被 NeurIPS 2020 接收为 Spotlight，代码已在 GitHub 上公开，提供 PyTorch 和 TensorFlow 实现。

在线函数近似：增量记忆表示的形式化

我们的第一个洞见是从离散时间转向连续时间设置，这通常更容易进行理论分析。我们提出了一个非常自然的问题：**给定一个连续函数（一维）f(t)，我们能否在所有时间 t 维护一个固定大小的表示 c(t)∈R^N，使得 c(t)最优地捕捉 f 从时间 0 到 t 的历史？**

然而，这个问题还没有完全定义——我们需要明确：

1. 近似质量：对函数历史的"最优近似"是什么？我们需要指定一个度量（或权重函数）来告诉我们对过去的每个时间点关心程度。

2. 基底：我们如何将连续函数压缩成固定长度的向量？我们可以将函数投影到 N 维子空间，并存储其在任意基底展开的 N 个系数。为简单起见，本文假设我们使用多项式基底。

直观地说，我们可以将记忆表示 c(t)∈R^N 视为 f(t)历史的最优多项式近似的系数向量。

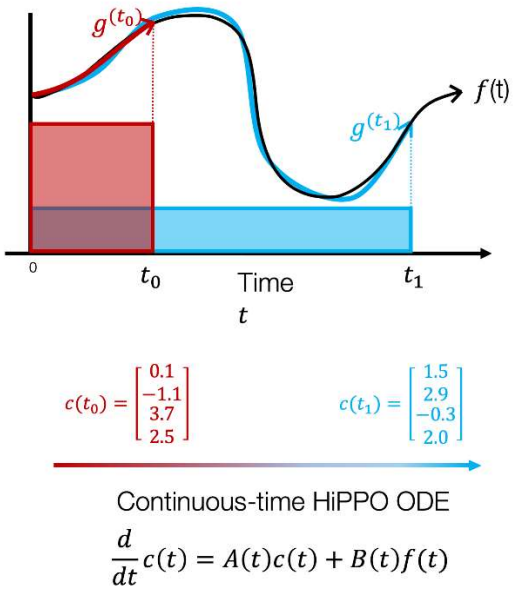
HiPPO 框架（高阶多项式投影算子）

注意，给定度量（并假设多项式基底），在线函数近似问题现在已完全指定！也就是说，对于任何输入函数 f(t)，所需的系数向量 c(t)（即我们期望的记忆表示）都已完全定义。剩下的问题是——我们如何计算它们？

HiPPO 框架将这个问题形式化，并提供了计算解的机制。尽管所需的系数 c(t)是作为近似问题的隐式解而抽象定义的，但令人惊讶的是，存在一个易于计算的闭式解。我们将技术细节留给完整论文，但值得一提的是，它们利用了经典的近似理论工具，如正交多项式。最终，解呈现为一个简单的线性微分方程形式，称为 HiPPO 算子：

$$C'(t) = A(t)c(t) + B(t)f(t)$$

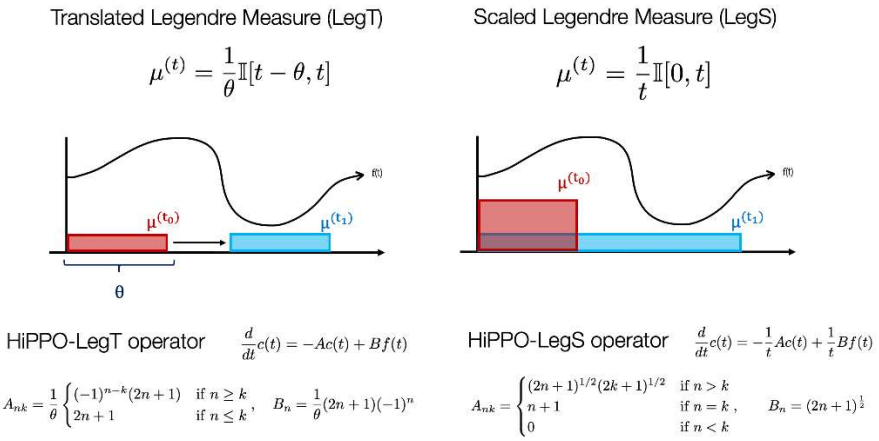
简而言之，HiPPO 框架接受一族度量，并给出具有闭式转移矩阵 A(t), B(t)的常微分方程。这些矩阵取决于度量，遵循这些动态可以找到系数 c(t)，根据度量最优地近似 f(t)的历史。



[图 1：HiPPO 框架。输入函数 f(t)（黑线）通过存储其根据指定度量（彩色框）的最优多项式投影（彩色线）系数被持续近似。这些系数随时间演化（红色，蓝色），遵循线性动力系统。]

HiPPO 的实例化

图 2 展示了 HiPPO 的一些具体例子。我们展示了两个基于均匀度量最简单的度量族。左侧的平移勒让德度量使用固定长度的滑动窗口；换句话说，它关注最近的历史。另一方面，缩放勒让德度量对当前时间之前的整个历史均匀加权。在这两种情况下，HiPPO 框架都为相应的常微分方程产生了封闭形式的公式（完整起见，展示了转移矩阵，它们实际上相当简单）。



[图 2. 简单度量及其对应 HiPPO 算子的示例。平移勒让德度量对过去的θ（超参数）个时间单位均匀加权，而缩放勒让德度量对所有历史均匀加权。]

(勒让德度量在 HiPPO 中扮演着定义"如何记忆"的关键角色。这些度量决定了如何对过去的信息进行加权和压缩。使用固定的框架提供了一个通用的、与任务无关的记忆机制。这种通用性使得模型可以适应各种不同的时间序列，而不需为每种序列类型设计专门的结构。
使用固定的勒让德框架可能看起来有些"一刀切"，但它实际上是一种精心设计的平衡方法。

它在通用性、理论保证、计算效率和灵活性之间取得了平衡。这种方法不是随意的，而是基于深入的数学和理论考虑，目的是创造一个既强大又通用的记忆机制。在实践中，它的效果已经被证明是非常有效的，特别是在处理长期依赖性方面。
虽然基本框架是固定的，但它可以与特定任务的学习结合使用。模型的其他部分（如神经网络层）可以学习如何最好地利用这种通用记忆结构。大脑中的记忆机制可能也使用类似的通用结构来处理不同类型的信息。译注)

从连续时间到离散时间

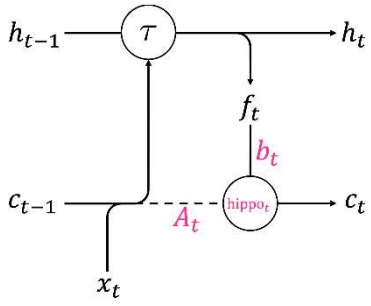
还有一个称为离散化的细节。通过使用标准技术来近似动力系统的演化，连续时间 HiPPO 常微分方程可以转换为离散时间线性递推。此外，这一步允许 HiPPO 灵活地处理不规则采样或缺失数据：只需根据给定的时间戳演化系统即可。

对实践者而言：**HiPPO 实现为简单的线性递推 c_(t+1) = A_t c_t + B_t f_t，其中 c_t 是当前的状态向量，f_t 是当前的输入; A_t 和 B_t 是由勒让德矩阵定义的转移矩阵，它们都有封闭形式的公式。***HiPPO 的核心就是这样一个线性递推*! (勒让德多项式形成一个正交基，这使得状态表示更加紧凑和无冗余--意味着它们可以高效地表示各种函数。勒让德矩阵的结构允许模型有效地捕捉长期依赖关系--不同阶数的多项式可以表示不同时间尺度的信息。HiPPO 在理论上可以以近乎最优的方式压缩和表示历史信息。译注)

整合到机器学习模型中

从本质上讲，HiPPO 是一个简单的线性递推，可以以多种方式集成到端到端模型中。我们关注循环神经网络（RNN），因为它们与涉及随时间演化的状态的动力系统有联系，就像 HiPPO 一样。HiPPO-RNN 是执行此集成的最简单方式：

$$c_{t+1} = A_t c_t + B_t f_t$$



[图 3. （上）HiPPO 在离散序列上具有简单线性递推的形式。（下）HiPPO-RNN 单元图。]

1. 从标准 RNN 递推 h_t = τ(h_(t-1), x_t)开始，该递推通过任何非线性函数τ给定输入 x_t 来演化隐藏状态 h_t
2. 将状态投影到较低维度的特征 f_t
3. **使用 HiPPO 递推创建 f_t 历史的表示 c_t**，并将这个表示反馈回τ中

(直接替代τ? 实际上, HiPPO 的核心是一个线性递推，这与大多数 RNN 模型的结构都兼容。译注)

HiPPO-RNN 的特殊情况

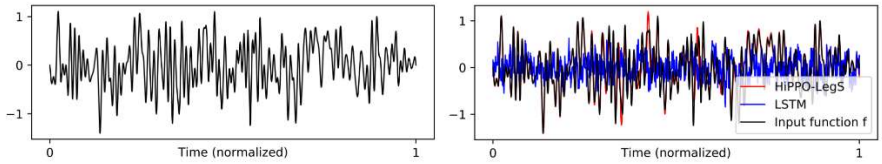
熟悉 RNN 的读者可能会注意到，这看起来与 LSTM 等其他模型的单元图非常相似。事实上，几个常见模型与之密切相关：

1. 最流行的 RNN 模型是 LSTM 和 GRU，它们依赖于门控机制。特别是，LSTM 的单元状态执行递推 c_(t+1) = α_t c_t + β_t f_t, 其中α_t, β被称为"遗忘"和"输入"门。注意与 HiPPO 递推 c_(t+1) = A_t c_t + B_t f_t 的相似性。事实上，**这些门控 RNN 可以被视为 HiPPO 的特殊情况，具有低阶（N=1）近似和输入依赖的离散化**！因此，HiPPO 为这些流行模型提供了新的见解，并展示了如何推导出最初作为启发式引入的门控机制。

2. HiPPO-LegT 模型，即平移勒让德度量的 HiPPO 实例化，与最近提出的勒让德记忆单元（LMU）完全等价。我们的证明也更简短，只需遵循 HiPPO 框架的步骤即可！

Hippo 从不忘记

让我们看看这些模型在基准测试中的表现如何。首先，我们测试 HiPPO 是否解决了它被设计来解决的问题——在线函数近似。图 4 显示，它可以**以良好的保真度近似一百万个时间步的序列**。请记住，这在使用有限的隐藏单元在线处理函数时是有效的；它可以在任何时间点重建函数的局部。



[图 4. （左）带限白噪声函数，采样为长度为 1000000 的序列。（右）使用 256 个隐藏单元处理序列后的近似函数重建。HiPPO 紧密匹配原始函数（MSE 0.02），而 LSTM 产生随机噪声（MSE 0.25）。]

其次，我们在标准的 Permuted MNIST 基准测试上进行测试，其中模型必须一次处理一个像素的输入图像，并在消耗整个序列后输出分类。这是测试序列模型长期依赖性的经典基准，因为它们必须记住近 1000 个时间步前的输入。我们 HiPPO 框架的多个实例化，包括上述描述的 HiPPO-LegS 和 HiPPO-LegT 模型，显著超越了其他循环模型，达到了 98.3%的测试准确率，而之前的最佳结果为 97.15%。事实上，它们甚至超过了使用全局上下文的非循环序列模型，如扩张卷积和变换器。完整结果可在表 4+5 中找到。

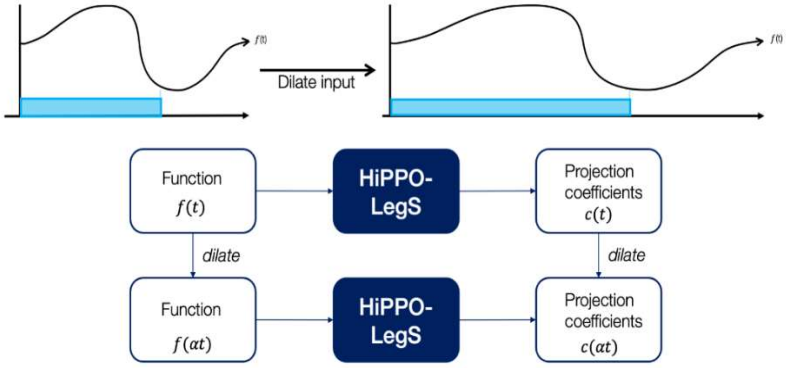
HiPPO-LegS 的时间尺度鲁棒性

最后，我们将探讨我们最有趣的模型——对应于缩放勒让德（LegS）度量的模型的一些理论性质。作为动机，细心的读者可能会想：HiPPO 的不同实例化之间有什么区别？度量如何影响模型？以下是度量的直观解释如何转化为下游 HiPPO 模型的理论性质的一些例子：

1. 梯度界限：由于这个度量表明我们关心整个过去，信息应该能够很好地通过时间传播。事实上，我们证明了**模型的梯度范数随时间呈多项式衰减，而不是指数衰减（即普通 RNN 的梯度消失问题）**。

2. 计算效率：转移矩阵 A_t 实际上具有特殊结构，递推可以在线性而不是二次时间内计算。我们假设这些效率属性普遍成立（即对所有度量成立），并且更广泛地与正交多项式及其相关计算的效率有关（例如，SODA 论文）。

3. 时间尺度鲁棒性：最有趣的是，缩放度量不受输入函数演化速度的影响；图 5 直观地说明了 HiPPO-LegS 如何具有扩张等变性。



[图 5.（上）由于缩放勒让德度量随时间拉伸，直观上，扩张输入函数不应改变投影。（下）说明 HiPPO-LegS 算子对时间扩张等变的交换图。]

Generalization	LSTM	GRU-D	ODE-RNN	NCDE	LMU	HiPPO-LegS
100Hz -> 200Hz	25.4	23.1	41.8	44.7	6.0	**88.8**
200Hz -> 100Hz	64.6	25.5	31.5	11.3	13.1	**90.1**

表格显示了在训练和测试序列之间存在分布偏移的轨迹分类数据集上的结果（即，由于部署时时间序列以不同速率采样而产生）；HiPPO 是唯一能够泛化到新时间尺度的方法！

结论

1. 通过提出和解决连续时间形式化问题，可以解决维护序列数据记忆表示的问题。
2. HiPPO 框架解释了几个先前的序列模型，同时产生了具有新颖性质的新模型。
3. 这只是冰山一角——HiPPO 有许多技术扩展、与其他序列模型的丰富联系，以及待探索的潜在应用！

尝试一下

HiPPO 的 PyTorch 和 TensorFlow 代码可在 GitHub 上获得，其中 HiPPO-RNN 可以作为大多数基于 RNN 的模型的直接替代品。文中提到的 HiPPO 实例化及更多实例化的闭式公式和实现都已提供。更多详情，请参阅完整论文。

脚注

1. 度量在函数空间上诱导了希尔伯特空间结构，因此存在唯一的最优近似——投影到所需的子空间。↩
2. 著名的正交多项式的例子包括切比雪夫多项式和勒让德多项式。我们的方法名称，如 LegS（缩放勒让德），基于与其度量对应的正交多项式族。↩
3. 我们将 HiPPO 递推集成到 RNN 中的方式略有不同，因此 HiPPO-LegT 和勒让德记忆单元（LMU）的完整 RNN 版本略有不同，但核心线性递推是相同的。↩