State Space Models using HiPPO

Sanidhya Kaushik **Sathvik Manthri** **Naman Mishra** **Deep Birenbhai Vaghasiya**

1 Introduction

When learning from sequential data, a challenge is to store long-term information efficiently. HiPPO (high-order polynomial projection operators) aims to solve this by approximating using a polynomial basis.

We assume the input signal to be continuous $\mathbb{R} \to \mathbb{R}$, even though it is received discretely. We fix a (time-varying) 'measure', which specifies the importance of each point in the signal. The given measure $\mu^{(t)}$ defines the inner product

$$\langle f, g \rangle_{\mu^{(t)}} = \int f(x)g(x)\mathrm{d}\mu^{(t)}(x). \tag{1}$$

For smooth measures, we can write $\mathrm{d}\mu^{(t)}(x) = \omega(t, x)\mathrm{d}x$. We want a polynomial $g^{(t)}$ that minimizes the corresponding L^2 error

$$\|f - g^{(t)}\|^2_{\mu^{(t)}} = \int (f(x) - g^{(t)}(x))^2 \omega(t, x)\mathrm{d}x. \tag{2}$$

The paper primarily considers three measures. We give a memory-based interpretation for each of these.

LegT. The translated Legendre measure, parameterized by θ.

$$\omega(t, x) = \frac{1}{\theta}\mathbf{1}_{[t-\theta,t]}(x).$$

This attempts to remember only a fixed duration into the past, giving equal importance to each instance.

LagT. The translated Laguerre measure.

$$\omega(t, x) = e^{x-t}\mathbf{1}_{[-\infty,t]}(x).$$

This attempts to remember the entire history, giving far higher importance to recent events.

LegS. The scaled Legendre measure.

$$\omega(t, x) = \frac{1}{t}\mathbf{1}_{[0,t]}(x).$$

This attempts to remember the entire history, but giving equal importance to every instance.

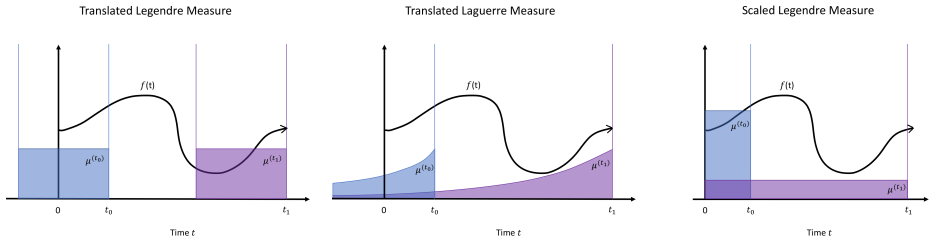

Figure 1: Illustration of LegT, LagT and LegS measures

These are illustrated in Fig. 1. Besides these, The paper also considers some Fourier and Chebyshev measures in passing.

Preprint. Under review.

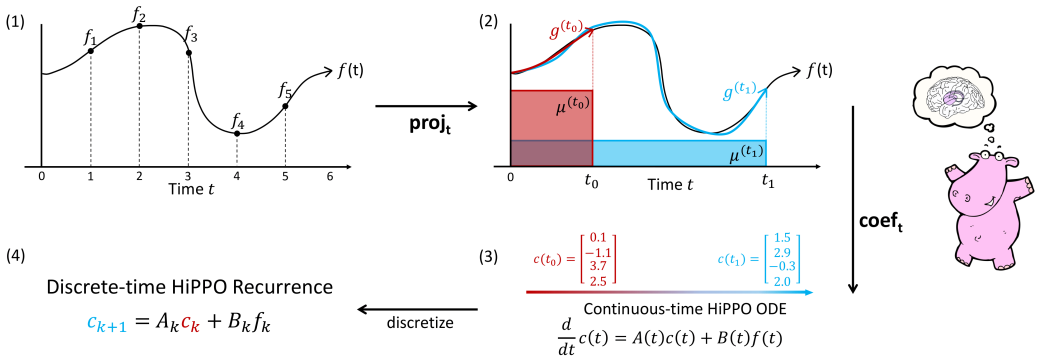

Figure 2: The HiPPO operator

2 Key idea

The key idea is to store the *coefficients* of the best polynomial approximation in some basis, and study how they evolve through time. The authors define for this the "hippo" operator, which takes a function as its input and outputs the coefficients of the best polynomial approximation.

Orthogonal polynomials are an obvious choice for working with coefficients, since the coefficients can be computed using inner products. Specifically, the optimal polynomial approximation is given by

$$g^{(t)}(x) = \sum_{i=0}^{n} \langle f,\ P_i^{(t)} \rangle_{\mu^{(t)}} P_i^{(t)}(x) =: \sum_{i=0}^{n} c_i(t) P_i^{(t)}(x). \tag{3}$$

where $(P_i^{(t)})_{i=0}^n$ is an orthogonal basis with respect to $\mu^{(t)}$.

Each coefficient $c_i(t)$ is computed as

$$c_i(t) = \int f(x) P_i^{(t)}(x) \omega(t,x) \mathrm{d}x. \tag{4}$$

Differentiating this with respect to time, we get

$$\frac{\mathrm{d}}{\mathrm{d}t} c_i(t) = \int f(x) \frac{\partial}{\partial t} P_i^{(t)}(x) \omega(t,x) \mathrm{d}x + \int f(x) P_i^{(t)}(x) \frac{\partial}{\partial t} \omega(t,x) \mathrm{d}x. \tag{5}$$

The derivatives $\frac{\partial}{\partial t} P_i^{(t)}$ are polynomials of degree $i-1$, so they can be written as a linear combination of $\{P_j\}_{j=0}^i$. Thus the first term is a linear combination of terms of the form

$$\int f(x) P_j^{(t)}(x) \omega(t,x) \mathrm{d}x$$

which are precisely the coefficients $c_j(t)$.

If $\frac{\partial}{\partial t}\omega(t,x)$ can also be expressed in terms of ω in a 'nice' way, then we can write the second term also using $c(t)$ and $f(t)$. For the measures dealt with in the paper, this is always of the form

$$\frac{\partial}{\partial t}\omega(t,x) = \varphi(t)(\omega(t,x) + \text{ some combination of Dirac-}\delta\text{ functions})$$

for some function $\varphi(t)$. [1] The delta functions integrate out to give f evaluated at some points. The $\omega(t,x)$ term integrates to give

$$\int f(x) P_i^{(t)}(x) \varphi(t) \omega(t,x) \mathrm{d}x = \varphi(t) c_i(t).$$

[1] For measures that don't follow this, they are 'tilted' by dividing by a suitably chosen function χ, such that ω/χ then satisfies the required property.

Measure	Train accuracy	Test accuracy	Validation accuracy
LegT	96.9%	96.7%	96.1%
LagT	78.0%	79.1%	78.9%
LegS	98.6%	96.4%	96.1%

Table 1: Experiment results

This gives a linear ODE for the coefficients $c(t)$.

$$\frac{\mathrm{d}}{\mathrm{d}t}c(t) = A(t)c(t) + B(t)f(t). \tag{6}$$

It turns out that for the measures considered in the paper, A and B are independent of t, so that

$$\frac{\mathrm{d}}{\mathrm{d}t}c(t) = Ac(t) + Bf(t). \tag{7}$$

The ODE obtained can be discretized using standard techniques such as the Euler and bilinear methods.

$$c_{k+1} = Ac_k + Bf_k$$

The following results are derived in the paper.

Theorem 1 *For LegT and LagT, the coefficients of the best polynomial approximation are given by linear time-invariant (LTI) ODEs* $\frac{\mathrm{d}}{\mathrm{d}t}c(t) = -Ac(t) + Bf(t)$*, where* $A \in \mathbb{R}^{N\times N}, B \in \mathbb{R}^{N\times 1}$*:*

LegT:

$$A_{nk} = \frac{1}{\theta}\begin{cases}(-1)^{n-k}(2n+1) & \text{if } n \geq k\\ 2n+1 & \text{if } n \leq k\end{cases}, \qquad B_n = \frac{1}{\theta}(2n+1)(-1)^n \tag{8}$$

LagT:

$$A_{nk} = \begin{cases}1 & \text{if } n \geq k\\ 0 & \text{if } n < k\end{cases}, \qquad B_n = 1 \tag{9}$$

Theorem 2 *The continuous-* (10) *and discrete-* (11) *time dynamics for* ***HiPPO-LegS*** *are:*

$$\frac{\mathrm{d}}{\mathrm{d}t}c(t) = -\frac{1}{t}Ac(t) + \frac{1}{t}Bf(t) \tag{10}$$

$$c_{k+1} - c_k = -\frac{A}{k}c_k + \frac{1}{k}Bf_k \tag{11}$$

$$A_{nk} = \begin{cases}(2n+1)^{1/2}(2k+1)^{1/2} & \text{if } n > k\\ n+1 & \text{if } n = k\\ 0 & \text{if } n < k\end{cases}, \qquad B_n = (2n+1)^{\frac{1}{2}}$$

3 Why is this news?

What makes HiPPO stand out is that it operates on sequential data, but its update equation can be unrolled to get

$$c_k = A^k Bf_0 + A^{k-1}Bf_1 + \cdots + Bf_k = K * f,$$

where $K = (A^k B, \ldots, AB, B)$ and $f = (f_0, \ldots, f_k)$. We incorporate HiPPO into a neural network. While training, the model can be trained similar to a CNN, but made to predict sequential data like an RNN.

4 Experiments

We have run the code from `https://github.com/HazyResearch/hippo-code` on the permuted-MNIST dataset to verify the paper's claims. We ran the code for 10 epochs with a hidden layer size (equal to the polynomial degree) of 256. Our results are recorded in Table 1.

References

[1] Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra and Christopher Ré. (2020) HiPPO: Recurrent Memory with Optimal Polynomial Projections. Available at: https://doi.org/10.48550/arXiv.2110.13985.

[2] Albert Gu, Karan Goel and Christopher Ré. (2021) Efficiently Modeling Long Sequences with Structured State Spaces. Available at: https://doi.org/10.48550/arXiv.2111.00396

[3] Sasha Rush and Sidd Karamcheti. The Annotated S4 v3. Available at: https://srush.github.io/annotated-s4