Data-Driven Lipschitz Models

Edoardo Manino[1] Iury Bessa[2] Andevaldo Vitorio[2]

[1]The University of Manchester, UK
[2]Universidade Federal do Amazonas, Brazil

1 Preliminaries

1.1 Lipschitz Continuity

A function g is Lipschitz-continuous in norm p with Lipschitz constant $c = \text{Lip}_p(g)$ if the following holds:

$$||g(x) - g(y)||_p \leq c||x - y||_p \tag{1}$$

for any $x, y \in \mathcal{B}$ with $\mathcal{B} \subseteq \mathbb{R}^d$. For example, a ball around $x = 0$.

1.2 Lipschitz-Bounded Neural Networks

Assume that we have access to a class of estimators $\hat{f} \in \mathcal{F}(p, c)$ such that $\text{Lip}_p(\hat{f}) = c$. An instance of such class are Lipschitz-Bounded Neural Networks, which typically provide any c for $p = 2$ [1] or more rarely $p = \infty$ [3].

1.3 Problem Statement

Given a system $\dot{x} = f(x)$, approximate it as follows:

$$\dot{x} = \hat{f}(x) + \Delta(x) \tag{2}$$

where the model $\hat{f}(x)$ was trained on dataset $\mathcal{T}\{(x, \dot{x})_i\}$ and $\Delta(x)$ is a conical function such that:

$$\big(\Delta(x) - \alpha x\big)^T \big(\Delta(x) - \beta x\big) \leq 0 \tag{3}$$

and $\Delta(x) \equiv f(x) - \hat{f}(x)$.

2 Methodology

The following work is in progress; some details might be missing.

2.1 Matrix-Valued Reparametrisation

In the following, we will reparametrise both the ground truth $f(x)$ and the model $\hat{f}$ as follows:

$$f(x) = G(x)x \tag{4}$$

where $G : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ is a matrix-valued function of the input. As a result, the approximation error $\Delta(x)$ takes the following form:

$$\begin{aligned} \Delta(x) &= f(x) - \hat{f}(x) \\ &= \big(G(x) - \hat{G}(x)\big)x \\ &= \Delta_g(x)x \end{aligned} \tag{5}$$

where $\Delta_g(x) \equiv G(x) - \hat{G}(x)$.

2.2 Conical Error Bounds

The reparametrisation above allows us to derive a simple form for the constant α and β in the conical constraint of Equation 3. More specifically, we are going to set:

$$\alpha = -\beta = \max_{x\in\mathcal{B}} ||\Delta_g(x)||_2 = \Delta_{\mathcal{B}} \tag{6}$$

where $\mathcal{B}$ is an arbitrary region where we want the conical constraint to be valid. We can show that this is indeed the case for all $x \in \mathcal{B}$, since:[1]

$$\begin{aligned} &\big(\Delta(x) - \alpha x\big)^T \big(\Delta(x) - \beta x\big) \\ &\quad= \big(\Delta_g(x)x - \Delta_{\mathcal{B}}x\big)^T \big(\Delta_g(x)x - \Delta_{\mathcal{B}}x\big) \\ &\quad= x^T\big(\Delta_g(x) - \Delta_{\mathcal{B}}I\big)^T \big(\Delta_g(x) - \Delta_{\mathcal{B}}I\big)x \\ &\quad= x^T\Delta_g(x)^T\Delta_g(x)x - \Delta_{\mathcal{B}}^2 x^Tx + \Delta_{\mathcal{B}}x^T\Big(\Delta_g(x)^T - \Delta_g(x)\Big)x \\ &\quad= ||\Delta_g(x)x||_2^2 - \Delta_{\mathcal{B}}^2||x||_2^2 + 0 \\ &\quad\le ||\Delta_g(x)||_2^2||x||_2^2 - \Delta_{\mathcal{B}}^2||x||_2^2 \\ &\quad= \Big(||\Delta_g(x)||_2^2 - \Delta_{\mathcal{B}}^2\Big)||x||_2^2 \\ &\quad\le 0, \end{aligned} \tag{7}$$

where the third term of the addition is zero since $\Delta_g(x)^T - \Delta_g(x)$ is a skew-symmetric matrix,[2] the first inequality is a consequence of the sub-multiplicativity of the matrix norm (Cauchy–Schwarz inequality), and the second inequality is a consequence of the definition of $\Delta_{\mathcal{B}}$ in Equation 6.

[1] A simpler proof should be possible by setting $\beta = -\alpha$ in the first equation.

[2] If S is skew-symmetrix, then $S^T = -S$ and $x^TSx = (x^TSx)^T = x^TS^Tx = -x^TSx = 0$.

2.3 Conical Bound Decomposition

For any estimator $\hat{G}$ of the reparametrised ground truth G, we can further bound the threshold in Equation 6 from above. Crucially, we can decompose such bound into three meaningful components, as shown below. To do so, we introduce the auxiliary function:

$$t(x) = \arg\min_{x' \in \mathcal{T}} ||x' - x||_2 \tag{8}$$

which returns the training input x' that is nearest to the arbitrary input x. With it, we can write the following:

$$\begin{aligned}
\Delta_{\mathcal{B}} &= \max_{x \in \mathcal{B}} ||\Delta_g(x)||_2 \\
&= \max_{x \in \mathcal{B}} ||G(x) - \hat{G}(x)||_2 \\
&= \max_{x \in \mathcal{B}} ||G(x) - G(t(x)) + G(t(x)) - \hat{G}(t(x)) + \hat{G}(t(x)) - \hat{G}(x)||_2 \\
&\leq \max_{x \in \mathcal{B}} ||G(x) - G(t(x))||_2 \\
&\qquad + \max_{x \in \mathcal{B}} ||G(t(x)) - \hat{G}(t(x))||_2 \\
&\qquad + \max_{x \in \mathcal{B}} ||\hat{G}(t(x)) - \hat{G}(x)||_2 \\
&\leq (c_G + c_{\hat{G}}) \max_{x \in \mathcal{B}} ||x - t(x)||_2 + \max_{x \in \mathcal{B}} ||G(t(x)) - \hat{G}(t(x))||_2
\end{aligned} \tag{9}$$

where the first inequality is a consequence of applying both $||A + B||_2 \leq ||A||_2 + ||B||_2$ and $\max_x \{u(x) + v(x)\} \leq \max_x u(x) + \max_x v(x)$, while the second inequality comes from the assumption that both the ground truth G and our estimator $\hat{G}$ are Lipschitz with constants c_G and $c_{\hat{G}}$ respectively.

Note that these components quantify three different sources of error:

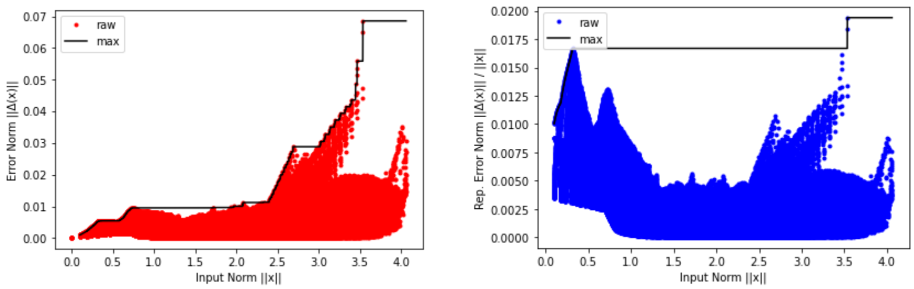
- c_G, the smoothness of the ground-truth function G;
- $c_{\hat{G}}$, the smoothness of the estimator $\hat{G}$;
- $G(t(x)) - \hat{G}(t(x))$, the error on the training set $\mathcal{T}$.

Unfortunately, we only have access to the second, since $c_{\hat{G}}$ is set during training. In contrast, computing c_G and $G(t(x)) - \hat{G}(t(x))$ requires direct access to the ground-truth function G, which we do not have. For the time being we will ignore this limitation.[3]

3 Example

Here, we quantify values of the bound in Equation 9 on a practical example.

[3] I am open to suggestions on how to bypass this issue.



(a) the maximum training error $||\Delta(x)||_2$ correlates with the input ball $||x||_2 \leq \mathcal{B}$.

(b) for any $||x||_2 \geq 10^{-6}$, we compute the lower bound $||\Delta_g(x)||_2 \geq ||\Delta(x)||_2/||x||_2$.

Figure 1: empirical error on training inputs $x \in \mathcal{T}$.

3.1 Van der Pol Oscillator

Ground-truth model:

$$\begin{aligned} \dot{x}_1 &= \mu\left(x_1 - \frac{1}{3}x_1^3 - x_2\right) \\ \dot{x}_2 &= \frac{1}{\mu}x_1 \end{aligned} \tag{10}$$

where we set $\mu = 1$ in our experiments.

3.2 Experimental Setup

In no particular order:

- Our Lipschitz-Bounded Neural Network $\hat{f}$ is based on an Almost-Orthogonal Layer [2]. The architecture has 4 layers with 8 neurons each. We fix $\text{Lip}_p(\hat{f}) = 1$ with $p = 2$.
- Our training set contains multiple trajectories sampled from Equation 10. The starting conditions are in the interval $x_i \in [-2.5, +2.5]$ with grid size $\ell = 0.1$.
- Training: batch size 64, epochs 1, learning rate $1e^{-4}$, Adam optimiser.

3.3 Results

The (approximate) conical threshold we get is the following (see Equation 9):

$$\begin{aligned} \Delta_{\mathcal{B}} &\leq (c_G + c_{\hat{G}}) \max_{x \in \mathcal{B}} ||x - t(x)||_2 + \max_{x \in \mathcal{B}} ||G(t(x)) - \hat{G}(t(x))||_2 \\ &\approx (2.67 + 1)0.07 + 0.02 = 0.28 \end{aligned} \tag{11}$$

for the input ball $\mathcal{B} = \{x : ||x||_2 \leq 4\}$. Smaller values could be achieved by reducing the size of $\mathcal{B}$. We give details on our computation below.

3.3.1 Training Error

Equations 9 and 11 contain a term $||\Delta_g(x')||_2$ that depends on training set input $x' = t(x)$ only. Unfortunately, we do not have access to the value of G in the definition $\Delta_g(x') \equiv G(x') - \hat{G}(x')$. Without further assumptions, the best we can do is introduce a *lower* bound on the training error as follows:[4]

$$||\Delta_g(x')||_2 \geq ||\Delta(x')||_2 / ||x'||_2 \tag{12}$$

for any $||x'||_2 > 0$. This bound is a consequence of $||AB||_2 \leq ||A||_2||B||_2$ and $\Delta(x) = \Delta_g(x)x$.

Figure 1a shows the training error $||\Delta(x')||_2$ against the input norm $||x'||_2$, while Figure 1b shows the lower bound in Equation 12. Note that in both figures the maximum training error increases as the size of the input ball $\mathcal{B}$ increases. As a result, smaller balls $\mathcal{B}$ yield smaller values of the threshold $\Delta_{\mathcal{B}}$ in Equation 11.

3.3.2 Lipschitz Constant of G

Equations 9 and 11 require knowledge of the Lipschitz constant c_G. In a fully data-driven approach, this quantity is unknown. For the sake of this experiment, we assumed perfect knowledge of the reparametrised ground truth:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} \mu - \frac{\mu}{3}x_1^2 & -\mu \\ \frac{1}{\mu} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{13}$$

see Equation 10 for the non-reparametrised version. The Lipschitz constant of the 2×2 matrix-valued function G in Equation 13 is dominated by the term $-\frac{\mu}{3}x_1^2$ which has first derivative $-\frac{2\mu}{3}x_1$. For $\mu = 1$ and $||x||_2 \leq 4$, such derivative is bounded by $\frac{8}{3} \approx 2.67$ in absolute value as reported in Equation 11. Smaller sizes of $\mathcal{B}$ would yield smaller values of x_1, thus reducing the Lipschitz constant c_G.

3.3.3 Nearest Training Input

Finally, Equations 9 and 11 rely on identifying the nearest training point $t(x)$ for any input x. Then, their distance $||x - t(x)||_2$ is used to bound the smoothness of G and $\hat{G}$ in conjunction with their respective Lipschitz constants. For a training set generated from a regular grid in a d-dimensional space $x \in \mathbb{R}^d$, we can bound $||x - t(x)||_2$ by considering the maximum distance of x from any point in the regular grid. More specifically, assume that for any two inputs ix, x' n the training set $\mathcal{T}$, we have:

$$x(i) - x'(i) = k\ell \qquad \text{with} \qquad k \in \mathbb{Z} \tag{14}$$

where ℓ is the grid size and $x(i)$ is the i-th entry of x. In this case, the maximum distance to any training point is:

$$\max_x ||x - t(x)||_2 \leq \frac{\ell}{2}\sqrt{d} \tag{15}$$

[4] Note that we would like to have an *upper* bound instead.

as long as x belongs to the convex hull of the training set $\mathcal{T}$. In Equation 11, we have $d = 2$ and $\ell = 0.1$ as per the latest dataset we are using.

References

[1] B. Prach, F. Brau, G. Buttazzo, and C. H. Lampert. 1-lipschitz layers compared: Memory speed and certifiable robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24574–24583, June 2024.

[2] B. Prach and C. H. Lampert. Almost-orthogonal layers for efficient general-purpose lipschitz networks. In S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, editors, *Computer Vision – ECCV 2022*, pages 350–365, Cham, 2022. Springer Nature Switzerland.

[3] B. Zhang, D. Jiang, D. He, and L. Wang. Rethinking lipschitz neural networks and certified robustness: A boolean function perspective. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 19398–19413. Curran Associates, Inc., 2022.