ECE 586 BH: Interplay between Control and Machine Learning **Fall 2023**

Lecture 8

A Control Perspective on Certifiably Robust Neural Networks, Part II

Lecturer: Bin Hu, Date:09/14/2023

In the last lecture, we stated that the neural network layer $x_{k+1} = A_k x_k + B_k \sigma(C_k x_k + b_k)$ with σ being slope-restricted on $[0, 1]$ is guaranteed to be 1-Lipschitz if there exists a diagonal positive definite matrix Λ_k such that the following matrix inequality holds

$$\begin{bmatrix} A_k^\mathsf{T} A_k - I & A_k^\mathsf{T} B_k \\ B_k^\mathsf{T} A_k & B_k^\mathsf{T} B_k \end{bmatrix} + \begin{bmatrix} C_k & 0 \\ 0 & I \end{bmatrix}^\mathsf{T} \begin{bmatrix} 0 & \Lambda_k \\ \Lambda_k & -2\Lambda_k \end{bmatrix} \begin{bmatrix} C_k & 0 \\ 0 & I \end{bmatrix} \leq 0. \tag{8.1}$$

Therefore, designing 1-Lipschitz layers boils down to finding the right choice of $(A_k, B_k, C_k, \Lambda_k)$. In this lecture, we will apply (8.1) to design new 1-Lipschitz layers. We will also discuss how to generalize (8.1).

8.1 SDP-Based Lipschitz Layers

We will start from the following result which was originally established in [1].

Theorem 8.1. *For any weight matrix $W_k \in \mathbb{R}^{m\times n}$, if there exists a nonsingular diagonal matrix T_k such that $W_k^\mathsf{T} W_k \leq T_k$, then the two following statements hold true.*

1. *The mapping $g(x) = W_k T_k^{-\frac{1}{2}} x_k + b_k$ is 1-Lipschitz.*
2. *The mapping $h(x) = x_k - 2W_k T_k^{-1} \sigma(W_k^\mathsf{T} x + b_k)$ is 1-Lipschitz if σ is ReLU,* tanh *or sigmoid.*

The proof of the first statement is straightforward, since g is affine in x. We have

$$\|g(x_k') - g(x_k)\|^2 = \|W_k T_k^{-\frac{1}{2}}(x_k' - x_k)\|^2 = (x_k' - x_k)^\mathsf{T} T_k^{-\frac{1}{2}} W_k^\mathsf{T} W_k T_k^{-\frac{1}{2}}(x_k' - x_k).$$

Based on the condition $W_k^\mathsf{T} W_k \leq T_k$, we have

$$\|g(x_k') - g(x_k)\|^2 \leq (x_k' - x_k)^\mathsf{T} T_k^{-\frac{1}{2}} T_k T_k^{-\frac{1}{2}}(x_k' - x_k) = \|x_k' - x_k\|^2.$$

Therefore, Statement 1 holds as desired. Based on Statement 1, for any 1-Lipschitz activation functions (not necessarily being slope-restricted on $[0, 1]$), the feed-forward network layer $x_{k+1} = \sigma(W_k T_k^{-\frac{1}{2}} x_k + b_k)$ is 1-Lipschitz. As we discussed before, training such layers for deep networks may experience the gradient vanishing issue, which can be addressed using the residual layer from Statement 2. Next, we discuss the proof of Statement 2.

Statement 2 can be proved using the general condition (8.1). Specifically, we set $A_k = I$, $B_k = -2W_k T_k^{-1}$, and $C_k = W_k^\mathsf{T}$. Then we substitute this choice of (A_k, B_k, C_k) into (8.1), and obtain

$$\begin{bmatrix} 0 & -2W_k T_k^{-1} \\ -2T_k^{-1} W_k^\mathsf{T} & 4T_k^{-1} W_k^\mathsf{T} W_k T_k^{-1} \end{bmatrix} + \begin{bmatrix} 0 & W_k \Lambda_k \\ \Lambda_k W_k^\mathsf{T} & -2\Lambda_k \end{bmatrix} \leq 0.$$

which is feasible with this choice of $\Lambda_k = 2T_k^{-1}$. To verify this, we substitute $\Lambda_k = 2T_k^{-1}$ into the above condition, and the left side becomes

$$\begin{bmatrix} 0 & 0 \\ 0 & 4T_k^{-1} W_k^\mathsf{T} W_k T_k^{-1} - 4T_k^{-1} \end{bmatrix},$$

which is negative semidefinite due to the fact $W_k^\mathsf{T} W_k \leq T_k$. Therefore, (8.1) is feasible for the choice of $(A_k, B_k, C_k, \Lambda_k) = (I, -2W_k T_k^{-1}, W_k^\mathsf{T}, 2T_k^{-1})$, and the residual layer in Statement 2 is 1-Lipschitz.

Why is Theorem 8.1 useful? The condition $W_k^\mathsf{T} W_k \leq T_k$ is much simpler than the general condition (8.1). If we try to directly use (8.1) to come up new choices of (A_k, B_k, C_k), the coupling between these decision parameters can lead to complications. In contrast, if we use the simplified condition $W_k^\mathsf{T} W_k \leq T_k$, we can just choose T_k as a function of W_k, and it is much easier to come up good choices of T_k. Overall, we can claim that the condition $W_k^\mathsf{T} W_k \leq T_k$ is easier to solve analytically than the original general condition (8.1). For example, we can re-derive existing 1-Lipschitz layers via using Theorem 8.1 and some trivial linear algebra facts:

- Spectral normalization: We can choose $T_k = \|W_k\|^2 I \geq W_k^\mathsf{T} W_k$, and then apply Statement 1 to recover this technique.
- Orthogonal layer: We can choose $T_k = I$ and enforce the equality $W_k^\mathsf{T} W_k = T_k = I$. Based on Statement 1, the orthogonal layers are 1-Lipschitz.
- AOL: We can choose $T_k = \mathrm{diag}(\sum_{j=1}^n |W_k^\mathsf{T} W_k|_{ij})$, and then the matrix $(T_k - W_k^\mathsf{T} W_k)$ becomes a real symmetric diagonally dominant matrix with non-negative diagonal entries. It is well known that such a matrix has to be positive semidefinite. Hence we have $W_k^\mathsf{T} W_k \leq T_k$, and Statement 1 can be directly applied to recover AOL.
- CPL: We can choose $T_k = \|W_k\|^2 I \geq W_k^\mathsf{T} W_k$ (this is the same choice of T_k as used for spectral normalization) and apply Statement 2 to recover this residual layer.

Notice that T_k is diagonal, and hence T_k^{-1} can be easily calculated and viewed as some scaling matrix. Therefore, Theorem 8.1 is also easy-to-use from the computation perspective.

Some new network structures. From Theorem 8.1, any choice of T_k satisfying $W_k^\mathsf{T} W_k \leq T_k$ immediately leads to two types of 1-Lipschitz layers. For example, for the choice of $T_k = \|W_k\|^2 I$, Statement 1 leads to the spectral normalization method, and Statement 2 leads to CPL. Now obviously, we can use the choices of T_k for orthogonal layers and AOL to construct residual layer counterparts. This immediately leads to the following new 1-Lipschitz residual layer structures:

- $x_{k+1} = x_k - 2W_k \sigma(W_k^\mathsf{T} x_k + b_k)$ with $W_k^\mathsf{T} W_k = I$
- $x_{k+1} = x_k - 2W_k \operatorname{diag}(\sum_{j=1}^n |W_k^\mathsf{T} W_k|_{ij})^{-1} \sigma(W_k^\mathsf{T} x_k + b_k)$

In addition, one can further refine the choice of T_k and obtain even more expressive 1-Lipschitz residual layers. See [1] for more examples.

Open questions. It is still unclear what is the best choice of T_k for the purpose of constructing 1-Lipschitz layers. The current choices from [1] achieve good certified robustness results due to the scalability for the deep network case. However, it is possible that there are other SDP solutions which will lead to more expressive 1-Lipschitz layers. In general, it is also possible to directly construct new layer structures from (8.1). In Homework 2, you will see one such example which uses a different solution of (8.1) (which has nothing to do with Theorem 8.1) to achieve competitive certified robustness results on TinyImageNet (see [4] for detailed discussions).

8.2 End-to-End Analysis for Multi-Layer Networks

In this section, we discuss how to generalize (8.1) for multi-layer networks. For simplicity, consider a two-layer network satisfying $x_1 = A_0 x_0 + B_0 \sigma(C_0 x_0 + b_0)$, and $x_2 = A_1 x_1 + B_1 \sigma(C_1 x_1 + b_1)$. To show such a network is 1-Lipschitz, we can apply (8.1) to each layer and then use the chain rule. Is there a better way of doing things? As a matter of fact, we can formulate an end-to-end SDP to directly ensure $\|x_2' - x_2\| \leq \|x_0' - x_0\|$ without worrying about what is going on in the middle layer. In other words, we do not care about whether we have $\|x_1' - x_1\| \leq \|x_0' - x_0\|$. We can allow $\|x_1' - x_1\| > \|x_0' - x_0\|$ but need to ensure $\|x_2' - x_2\| \leq \|x_0' - x_0\|$ in the end. Such an end-to-end approach is less conservative than the naive chain rule approach.

Now we will reverse engineer an end-to-end SDP ensuring $\|x_2' - x_2\| \leq \|x_0' - x_0\|$. Denote $w_0 = \sigma(C_0 x_0 + b_0)$ and $w_1 = \sigma(C_1 x_1 + b_1)$. Suppose σ is slope-restricted over $[0, 1]$. This property ensures the following quadratic inequalities

$$\begin{bmatrix} C_0(x_0' - x_0) \\ w_0' - w_0 \end{bmatrix}^\mathsf{T} \begin{bmatrix} 0 & \Lambda_0 \\ \Lambda_0 & -2\Lambda_0 \end{bmatrix} \begin{bmatrix} C_0(x_0' - x_0) \\ w_0' - w_0 \end{bmatrix} \geq 0,$$
$$\begin{bmatrix} C_1(x_1' - x_1) \\ w_1' - w_1 \end{bmatrix}^\mathsf{T} \begin{bmatrix} 0 & \Lambda_1 \\ \Lambda_1 & -2\Lambda_1 \end{bmatrix} \begin{bmatrix} C_1(x_1' - x_1) \\ w_1' - w_1 \end{bmatrix} \geq 0.$$

Therefore, to ensure $\|x_2' - x_2\| \leq \|x_0' - x_0\|$, we only need some matrix inequality that can lead to

$$\|x_2' - x_2\|^2 - \|x_0' - x_0\|^2$$
$$+ \begin{bmatrix} C_0(x_0' - x_0) \\ w_0' - w_0 \end{bmatrix}^\mathsf{T} \begin{bmatrix} 0 & \Lambda_0 \\ \Lambda_0 & -2\Lambda_0 \end{bmatrix} \begin{bmatrix} C_0(x_0' - x_0) \\ w_0' - w_0 \end{bmatrix} + \begin{bmatrix} C_1(x_1' - x_1) \\ w_1' - w_1 \end{bmatrix}^\mathsf{T} \begin{bmatrix} 0 & \Lambda_1 \\ \Lambda_1 & -2\Lambda_1 \end{bmatrix} \begin{bmatrix} C_1(x_1' - x_1) \\ w_1' - w_1 \end{bmatrix} \leq 0$$

If we can rewrite all the four terms on the left side of the above inequality in the following form:

$$\begin{bmatrix} x_0' - x_0 \\ w_0' - w_0 \\ w_1' - w_1 \end{bmatrix}^\mathsf{T} M_i \begin{bmatrix} x_0' - x_0 \\ w_0' - w_0 \\ w_1' - w_1 \end{bmatrix}$$

Then a SDP condition $\sum_{i=1}^4 M_i \leq 0$ can ensure the above desired inequality. Now let's figure out M_i. Notice $x_2 = A_1(A_0x_0 + B_0w_0) + B_1w_1 = A_1A_0x_0 + A_1B_0w_0 + B_1w_1$. We have

$$\|x_2' - x_2\|^2 = \begin{bmatrix} x_0' - x_0 \\ w_0' - w_0 \\ w_1' - w_1 \end{bmatrix}^\mathsf{T} \begin{bmatrix} A_0^\mathsf{T} A_1^\mathsf{T} \\ B_0^\mathsf{T} A_1 \\ B_1^\mathsf{T} \end{bmatrix} \begin{bmatrix} A_1A_0 & A_1B_0 & B_1 \end{bmatrix} \begin{bmatrix} x_0' - x_0 \\ w_0' - w_0 \\ w_1' - w_1 \end{bmatrix}$$
$$-\|x_0' - x_0\|^2 = \begin{bmatrix} x_0' - x_0 \\ w_0' - w_0 \\ w_1' - w_1 \end{bmatrix}^\mathsf{T} \begin{bmatrix} -I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_0' - x_0 \\ w_0' - w_0 \\ w_1' - w_1 \end{bmatrix}$$

Obviously, we have

$$M_1 = \begin{bmatrix} A_0^\mathsf{T} A_1^\mathsf{T} \\ B_0^\mathsf{T} A_1 \\ B_1^\mathsf{T} \end{bmatrix} \begin{bmatrix} A_1A_0 & A_1B_0 & B_1 \end{bmatrix}$$
$$M_2 = \begin{bmatrix} -I & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Similarly, we can figure out M_3 and M_4:

$$M_3 = \begin{bmatrix} C_0 & 0 & 0 \\ 0 & I & 0 \end{bmatrix}^\mathsf{T} \begin{bmatrix} 0 & \Lambda_0 \\ \Lambda_0 & -2\Lambda_0 \end{bmatrix} \begin{bmatrix} C_0 & 0 & 0 \\ 0 & I & 0 \end{bmatrix}$$
$$M_4 = \begin{bmatrix} C_1A_0 & C_1B_0 & 0 \\ 0 & 0 & I \end{bmatrix}^\mathsf{T} \begin{bmatrix} 0 & \Lambda_1 \\ \Lambda_1 & -2\Lambda_1 \end{bmatrix} \begin{bmatrix} C_1A_0 & C_1B_0 & 0 \\ 0 & 0 & I \end{bmatrix}$$

With the above choice of (M_1, M_2, M_3, M_4), we can formulate the end-to-end SDP. If there exist diagonal positive definite matrices (Λ_0, Λ_1) such that $\sum_{i=1}^4 M_i \leq 0$, then we have $\|x_2' - x_2\| \leq \|x_0' - x_0\|$. Sometimes the resultant SDP condition can be further simplified.

Exercise. Can we simplify the SDP when the network is feed-forward, i.e. $A_1 = A_0 = 0$ and $B_1 = B_0 = I$? Compare what you get to [2, Theorem 2]. Are the results the same? (Hint: One can merge $M_3 + M_4$ as

$$
\begin{aligned}
M_3 + M_4 &= \begin{bmatrix} C_0 & 0 & 0 \\ C_1A_0 & C_1B_0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}^\mathsf{T} \begin{bmatrix} 0 & 0 & \Lambda_0 & 0 \\ 0 & 0 & 0 & \Lambda_1 \\ \Lambda_0 & 0 & -2\Lambda_0 & 0 \\ 0 & \Lambda_1 & 0 & -2\Lambda_1 \end{bmatrix} \begin{bmatrix} C_0 & 0 & 0 \\ C_1A_0 & C_1B_0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix} \\
&= \begin{bmatrix} C_0 & 0 & 0 \\ C_1A_0 & C_1B_0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}^\mathsf{T} \left(\begin{bmatrix} 0 & 1 \\ 1 & -2 \end{bmatrix} \otimes \begin{bmatrix} \Lambda_0 & 0 \\ 0 & \Lambda_1 \end{bmatrix} \right) \begin{bmatrix} C_0 & 0 & 0 \\ C_1A_0 & C_1B_0 & 0 \\ 0 & I & 0 \\ 0 & 0 & I \end{bmatrix}
\end{aligned}
$$

Then use $A_1 = A_0 = 0$ and $B_1 = B_0 = I$ to simplify the expressions.)

Networks with arbitrary depth. The above analysis can be generalized to neural networks with arbitrary depth. Suppose $x_{k+1} = A_k x_k + B_k \sigma(C_k x_k + b_k)$, and we want to show $\|x'_{k+1} - x_{k+1}\| \leq \|x'_0 - x_0\|$. Denote $w_k = \sigma(C_k x_k + b_k)$ for all k. Then we can express x_{k+1} as a linear combination of x_0 and $\{w_i\}_{i=0}^k$. Consequently, we can use a similar reverse engineering approach to derive an end-to-end SDP via manipulating terms in the following quadratic form:

$$
\begin{bmatrix} x'_0 - x_0 \\ w'_0 - w_0 \\ w'_1 - w_1 \\ \vdots \\ w'_k - w_k \end{bmatrix}^\mathsf{T} M_i \begin{bmatrix} x'_0 - x_0 \\ w'_0 - w_0 \\ w'_1 - w_1 \\ \vdots \\ w'_k - w_k \end{bmatrix}.
$$

A detailed derivation is skipped and left as an exercise problem (we will discuss this in class).

8.3 More Discussions

Finally, it is beneficial to briefly mention several other useful generalizations of (8.1). Notice (8.1) was derived for ℓ_2 perturbations. The argument can be modified to give a SDP condition for Lipschitz bounds under ℓ_∞ perturbations. See [5] for more details. The quadratic constraint approach can also be used to address implicit learning models such as deep equilibrium models. See [3] for such results.

Bibliography

[1] A. Araujo, A. J. Havens, B. Delattre, A. Allauzen, and B. Hu. A unified algebraic perspective on lipschitz neural networks. In *The Eleventh International Conference on Learning Representations*, 2023.

[2] M. Fazlyab, A. Robey, H. Hassani, M. Morari, and G. Pappas. Efficient and accurate estimation of lipschitz constants for deep neural networks. *Advances in Neural Information Processing Systems*, 32, 2019.

[3] M. Revay, R. Wang, and I. R. Manchester. Lipschitz bounded equilibrium networks. *arXiv preprint arXiv:2010.01732*, 2020.

[4] R. Wang and I. Manchester. Direct parameterization of lipschitz-bounded deep networks. In *International Conference on Machine Learning*, pages 36093–36110, 2023.

[5] Z. Wang, G. Prakriya, and S. Jha. A quantitative geometric approach to neural-network smoothness. In *Advances in Neural Information Processing Systems*, 2022.