Paper Reading Report: *LipsNet: A Smooth and Robust Neural Network with Adaptive Lipschitz Constant for High-Accuracy Optimal Control* [1]

Wenting Liu
2021533058

Xiuyuan Qi
2021533177

Xinhe Chen
2021533093

1. Review of Article

1.1 Problem Statement

- **Background**

Due to the excellent ability of neural networks to fit nonlinear functions, deep reinforcement learning is a powerful method for solving optimal control problems. However, RL training strategies often suffer from action fluctuation issues, leading to control action distortion, mechanical component wear, and safety hazards.

- **Mathematical Problem Definition**

The article discussed action fluctuation problem occurred in optimal control problems. The problem aims to find a function $f : \mathbb{R}^n \to \mathbb{R}^m (m \geq 1)$, whose output $f(x)$ correspond to the optimal action at each state. However, the output sequence $f(x_t + \sigma)$ (σ stands for disturbance noise) often fluctuate significantly, which is not desirable in real-world applications. In the article, the authors used the Lipschitz constant of f to measure the smoothness of the policy. Given a function $f : \mathbb{R}^n \to \mathbb{R}^m$, if there exists a constant $K > 0$ satisfies

$$\|f(x_1) - f(x_2)\| \leq K \|x_1 - x_2\|, \forall x_1, x_2 \in \mathbb{R}^n \tag{1}$$

then f is said to be Lipschitz continuous with constant K. The goal is to minimize the Lipschitz constant K and meanwhile maintain the accuracy of the optimal action.

1.2 Methodology

To address this issue, the authors proposed LipsNet, a neural network with adaptive Lipschitz constant. It contains a normal MLP part and an extra layer after MLP to constraint the Lipschitz constant of the network. Specifically, the extra layer is written as

$$f_{MGN}(x) = K(x)\frac{f(x)}{||\nabla_x f(x)|| + \epsilon} \tag{2}$$

where ϵ is a small positive constant.

In the improved version, LipsNet-L (Figure 1), where f is locally Lipschitz continuous, $K(x)$ is obtained by a a separate MLP with Softplus activation function. The extended loss function is defined as $\mathcal{L}' = \mathcal{L} + \lambda K(x)^2$. It is guaranteed that the modified network is differentiable and can be trained with a simple gradient-based method. It may also replace MLP in actor network like TD3.(Figure 2)

1.3 Conclusion and Contribution

In short, LipsNet brings MLP into MGN for reinforcement learning for the first time, and utilize Lipschitz constant as an variable with auto optimization. Local Lipschitz continuous is a guarantee of in-time performance.

This provides a new policy container with the following features:

- Smooth policy function: this promises the robustness regarding observation noise.
- Better performance: it can output actions fast and with proper magnitude.
- Portability: can combine with different algorithm.

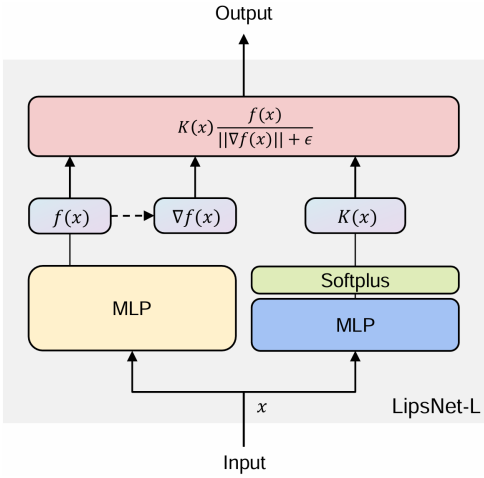

Figure 1: Structure of LipsNet-L. The input is x. The MLP in yellow represents $f(x)$. The MLP in blue with Softplus rep resents $K(x)$. The dashed line means derivative operation. The equation in pink represents MGN.

Algorithm 2 Update Parameters in LipsNet-L

Input: input vector x, loss function $\mathcal{L}$, parameter θ in network $f(x)$, parameter ϕ in network $K(x)$.

$f_{\mathrm{MGN}}(x) = K(x)\frac{f(x)}{\|\nabla_x f(x)\|+\epsilon}$ ▷ forward propagation

$\mathcal{L}' = \mathcal{L} + \lambda K(x)^2$

$\theta \leftarrow \theta - \eta_f \nabla_\theta \mathcal{L}'$ ▷ backward propagation

$\phi \leftarrow \phi - \eta_k \nabla_\phi \mathcal{L}'$ ▷ Lipschitz adjustment

Figure 2: Pseudocode of improved LipsNet, detailing forward and backward propagation during training. The modified loss function $\mathcal{L}'$ combines the original loss with a normalization term for $K(x)$.

2. Discussion of Key Points

2.1 Optimizing Lipschitz Constant K

The primary advantage of the article lies in its identification of a crucial issue in controlling fluctuations, namely, the emphasis on K-Lipschitz continuity in its analysis. This approach quantifies the degree of change in control actions over adjacent time units, which means $\|f(y) - f(x)\| = K\|y - x\|$ given $\|\nabla_x f(x)\| \leq K$.

2.2 Transformation of the Output Function

In the paper, the authors use equation 2 transformation neatly controls the output to a K-Lipschitz continuous result. By definition, the 2-norm of the Jacobian matrix of $f_{\mathrm{MGN}}(x)$ according to x is

$$\|\nabla_x f_{\mathrm{MGN}}(x)\| = K \cdot \left\| \frac{\nabla f}{\|\nabla f\| + \epsilon} \right\| = K \cdot \left\| \frac{\nabla f(\|\nabla f\| + \epsilon) - f(\nabla(\|\nabla f\| + \epsilon))^\top}{(\|\nabla f\| + \epsilon)^2} \right\|. \tag{3}$$

Because there are only piece-wise linear activation functions in f, its Jacobian matrix ∇f is a constant matrix. Thus, the norm of it, $\|\nabla f\|$, is a constant value. So $\nabla(\|\nabla f\| + \epsilon)$ is a zero vector.

Finally, the equation (3) can be simplified to

$$\|\nabla_x f_{\mathrm{MGN}}(x)\| = K \cdot \left\| \frac{\nabla f}{\|\nabla f\| + \epsilon} \right\| \leq K. \tag{4}$$

2.3 Use Activation Functions to Adjust the Output Range

Due to the generally limited range of action outputs in RL testing environments, the authors chose to use the tanh activation function to control the output range. This approach ensures monotonicity while maintaining the K-Lipschitz continuity of the final result. By definition, the 2-norm of the Jacobian matrix of $g(x)$ according to x is

$$\|\nabla_x g(x)\| = \|\nabla_{f_{MGN}(x)} g(x) \cdot \nabla_x f_{MGN}(x)\| \leq \|\nabla_{f_{MGN}(x)} g(x)\| \|\nabla_x f_{MGN}(x)\|. \tag{5}$$

Because `tanh` operates in element-wise and $0 < \nabla\mathtt{tanh} \leq 1$, $\nabla_{f_{MGN}(x)} g(x)$ is a diagonal matrix whose elements are all positive and smaller than 1. It implies that the norm $\|\nabla_{f_{MGN}(x)} g(x)\| \leq 1$. Therefore,

$$\|\nabla_x g(x)\| \leq 1 \cdot \|\nabla_x f_{MGN}(x)\| \leq K. \tag{6}$$

3. Extension Analysis

3.1 Personal Opinion: Physical Meaning

The authors addressed the mathematical validity of the formula but lacked an intuitive explanation. We will now discuss the physical meaning of the formula in Equation 2.

- **Action Accuracy**: The original output $f(x)$ represents the model's response to input x, providing corresponding action commands.
- **Acceleration Stability**: The gradient norm $\|\nabla_x f(x)\|$ measures the sensitivity of the model's output to input changes. A high gradient norm indicates sensitivity, potentially causing unstable actions. Normalizing the gradient norm ensures smooth and stable outputs, preventing instability from excessive sensitivity.
- **Balance Mechanism**: The factor $K(x)$ adjusts the output amplitude based on input conditions, balancing action accuracy and stability. Adjusting $K(x)$ finds a balance between response speed and output smoothness.

3.2 Further Improvement: Quantifying Sensitivity and Balancing with Smoothness

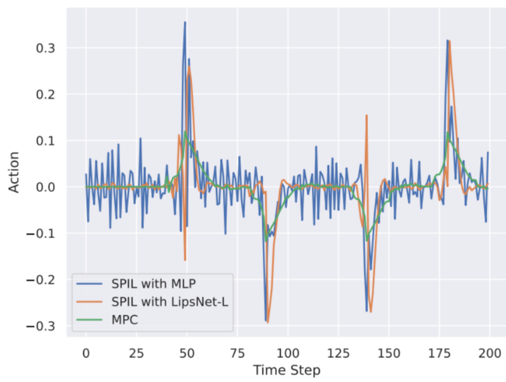

Figure 3.2 Results on vehicle trajectory tracking environment.

This figure 3.2 shows the results of the vehicle trajectory tracking experiment. The orange line (SPIL with MLP) reacts slower than the blue line (SPIL with LipsNet-L) and the green line (MPC). In practical applications like obstacle avoidance and cornering, controlling noise can reduce vehicle response sensitivity.

Our innovation is to quantify sensitivity and balance it with smoothness for an optimal tradeoff, setting a new standard for real-world applications. This approach enhances reaction sensitivity while maintaining smooth operations, better meeting practical demands.

- **Reaction Time Loss**

Suppose we are designing an autonomous driving model that needs to respond quickly when detecting an emergency turn signal. An emergency turn signal, such as a sudden sharp turn sign, requires the model to quickly adjust the steering wheel angle to avoid danger.

$$\text{Reaction Time Loss} = \int_0^T \left\| \frac{\partial f(x(t))}{\partial t} - k \cdot \frac{\partial x(t)}{\partial t} \right\|^2 dt \tag{7}$$

In this formula, $f(x(t))$ represents the output signal of the model, $\frac{\partial f(x(t))}{\partial t}$ denotes the rate of change of the model output with respect to time, and $\frac{\partial x(t)}{\partial t}$ represents the rate of change of the input signal with respect to time. The term k is a scaling factor used to match the units and scales of the two rates of change. The integral of the squared difference of these rates over the time interval $[0, T]$ measures the total loss in reaction time of the model.

- **Control gain loss**

Additionally, we define control gain loss to assess the deformation of the punishment:

$$\text{Control Gain Loss} = \left\| \int_0^T [f(x(t) - r(x(t))]\, dt \right\|^2 \tag{8}$$

In this formula, $f(x(t))$ represents the control signal of the model output, and $r(x(t))$ represents the control signal of the ground truth model output. The difference in the changes of their actions over the time interval T is represented by the integral.

Now we combine the reaction time loss and control gain loss to form a comprehensive sensitivity loss function:

$$\text{Sensitivity Loss} = \alpha \int_0^T \left\| \frac{\partial f(x(t))}{\partial t} - k \cdot \frac{\partial x(t)}{\partial t} \right\|^2 dt + \beta \left\| \int_0^T [f(x) - r(x)]\, dx \right\|^2 \tag{9}$$

where α and β are weighting parameters used to adjust the relative importance of the reaction time loss and control gain loss in the total loss function.

By combining these two loss functions,maybe we can design a model capable of responding quickly and accurately to emergency signals, such as brake signals.

References

[1] Xujie Song, Jingliang Duan, Wenxuan Wang, Shengbo Eben Li, Chen Chen, Bo Cheng, Bo Zhang, Junqing Wei, and Xiaoming Simon Wang. LipsNet: A smooth and robust neural network with adaptive Lipschitz constant for high accuracy optimal control. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 32253–32272. PMLR, 23–29 Jul 2023.