Dynamical System Inspired Adaptive Time Stepping Controller for ResNet Families

Yibo Yang, Jianlong Wu, Hongyang Li, Xia Li, Tiancheng Shen and Zhouchen Lin

◆ Summary

- Motivations

The connection between ResNet and dynamical system enables us to unravel the physics of residual networks using the rich theories and tools that are well-developed in numerical methods of ODEs. For ODE systems, adaptive method, such as Runge-Kutta-Fehlberg, is able to offer a good trade-off between the stability and efficiency. Can we also have an adaptive time stepping for ResNets to ensure both stability and performance?

- Contributions

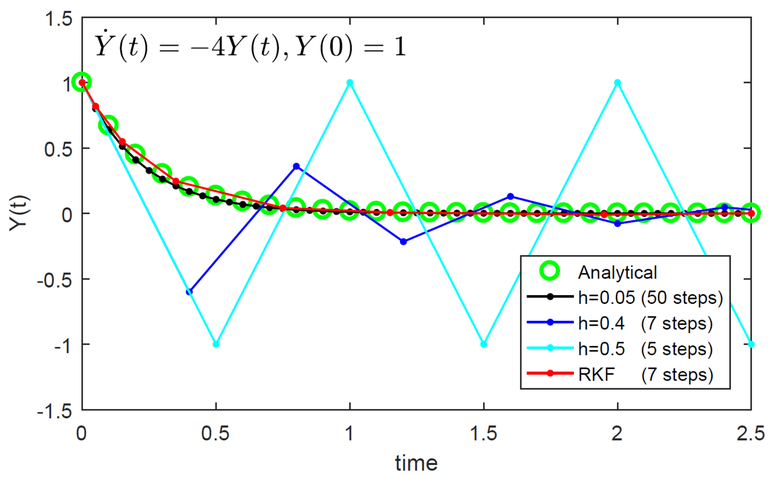
1. We analyze the correspondence between ODE and ResNet, establish a stability condition for ResNet, and point out the effects of step sizes on the stability and performance.
2. We develop an adaptive time stepping controller to optimize variable step sizes and evolution time jointly with the network training.
3. Experiments on ImageNet and CIFAR show that our method is able to improve both stability and accuracy. Besides, the improvements come with no additional cost in inference phase.

◆ Time Step for ODE

- The Runge-Kutta-Fehlberg method

$$|y_{j+1} - y(t_{j+1})| = O\left(\Delta t_j^{p+1}\right)$$

$$\Delta t_{j+1} = k \times \Delta t_j \times \left(\frac{Tol}{|y_{j+1} - \hat{y}_{j+1}|}\right)^{1/(p+1)}$$



where Δt_j is the current step size, k is a factor and Tol is a tolerance error. The method adaptively increases or reduces the next step size. It offers a stable solution, and is efficient for time steps.

◆ Time Step for ResNet

- Large steps cause instability

$$\|\mathbf{y}_D^{\epsilon} - \mathbf{y}_D\| \leq \epsilon \cdot \prod_{j=0}^{D-1} \left(1 + \|\mathbf{W}_j\|_2 \Delta t_j\right)$$

where $\|\mathbf{W}_j\|_2$ denotes the spectral norm of weight matrix in each residual block, ϵ is the initial perturbation that satisfies $\|\mathbf{y}_0^{\epsilon} - \mathbf{y}_0\| = \epsilon$.

- Small steps impede learning process

$$\frac{\partial L}{\partial \mathbf{y}_n} = \frac{\partial L}{\partial \mathbf{y}_D}\left[1 + \frac{\partial}{\partial \mathbf{y}_n}\sum_{i=n}^{D-1} \mathcal{F}(\mathbf{y}_i, \mathbf{W}_i)\Delta t_i\right]$$

◆ Adaptive Time Stepping Controller

- Variable step sizes and evolution time

$$\min_{\mathbf{w},\theta} J = \frac{1}{S}\sum_{s=1}^{S} \Phi(y_s(T), y_s^*) + \sum_{d=1}^{D-1} R\big(\mathbf{w}(t_d), \theta(t_d)\big),$$

$$s.t. \quad y_s(t_{d+1}) = y_s(t_d) + \mathcal{F}\big(y_s(t_d), \mathbf{w}(t_d)\big)\Delta t_d$$

$$\Delta t_d = \Theta\big(\mathbf{w}(t_d); \Delta t_1, \dots, \Delta t_{d-1}; \theta(t_d)\big)$$

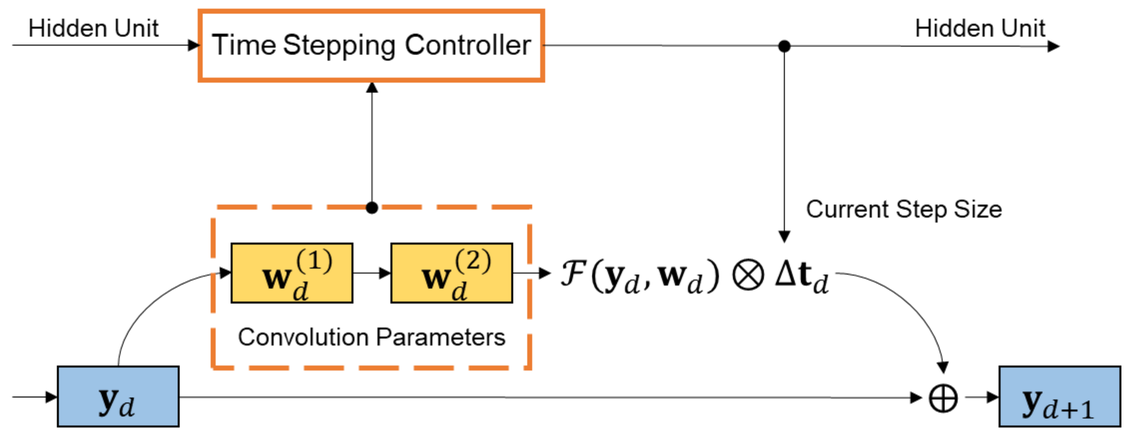
$$t_0 = 0, t_{d+1} = t_d + \Delta t_d, y_s(0) = y_s$$

$$T = t_D = \sum_{d=0}^{D-1} \Delta t_d, d = 0,1, \dots, D-1$$

where Φ is the loss function, R is the regularization, y_s^* is the label of input image y_s, S is the number of samples, and Θ is our introduced controller parameterized by $\{\theta_d\}_{d=1}^{D-1}$. This system has variable step sizes Δt_d and evolution time T.

- Dependent on weight parameters and previous steps

From our analyses, each step size should be aware of previous steps and the weight parameters in the current step. Our introduced controller connects different steps as an LSTM, and takes the parameters of each step as input, to decide the current step size. Because our controller is data-independent, the performance and stability gains bring little additional cost in inference.



◆ Selected Experimental Results

Tab. Top-1 error rates on ImageNet w/ and w/o our controller.

Models	Baseline			With Controller		
	Error(%)	Params (M)	GFLOPs	Error(%)	Params (M) (train / infer)	GFLOPs (train/infer)
ResNet-50	24.42	25.56	3.86	**23.63**	27.83 / 25.57	3.89 / 3.86
ResNeXt-50	22.84	25.03	3.77	**22.23**	27.30 / 25.04	3.80 / 3.77
SENet-50	23.27	28.09	3.87	**22.75**	30.36 / 28.10	3.90 / 3.87

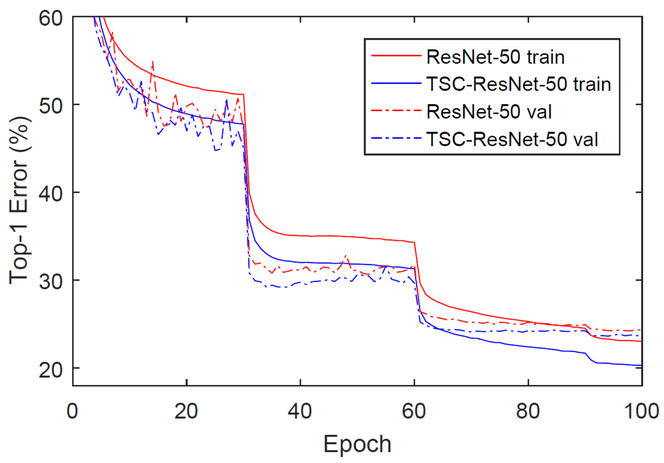


Fig. Top-1 error rate training curves w/ and w/o our controller.

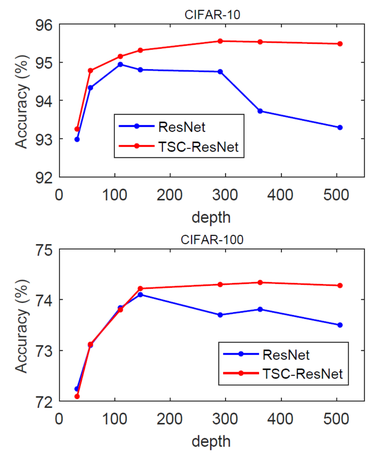


Fig. Stability with increasing depth.

Arxiv: https://arxiv.org/abs/1911.10305.pdf **Contact:** ibo@pku.edu.cn