Towards Robust ResNet

Jingfeng Zhang[1] Bo Han[2] Laura Wynter[3] Bryan Kian Hsiang Low[1] and Mohan Kankanhalli[1]

[1]Department of Computer Science, National University of Singapore

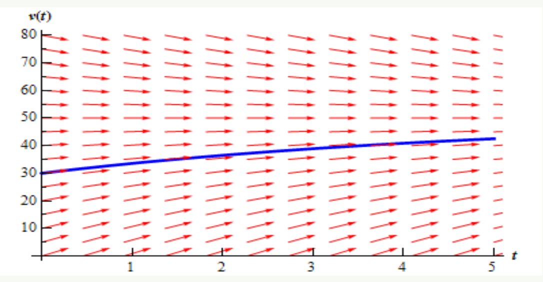
[2]RIKEN Center for Advanced Intelligence Project

[3]IBM Research, Singapore

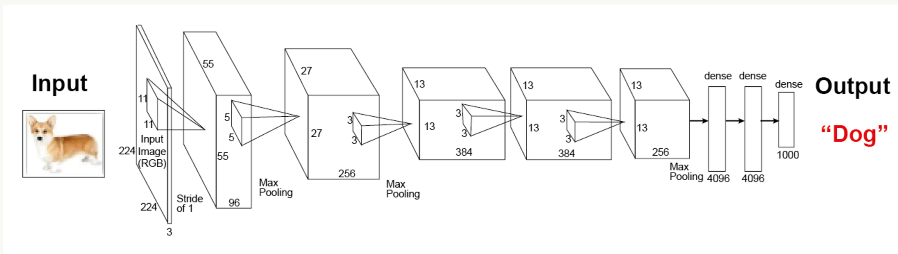
http://www.comp.nus.edu.sg/~lowkh/research.html

Motivation & Idea

Partial Differential Equation



Deep Neural Network



Object trajectory over time

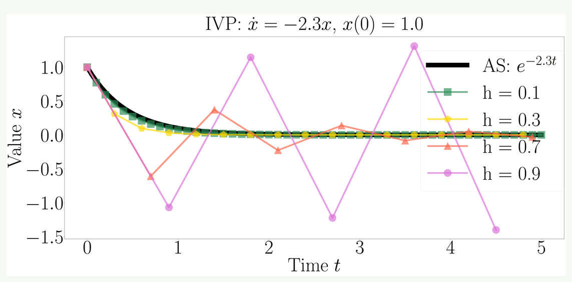
Feature transformation over depth of layers

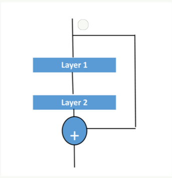
Explicit Euler Method

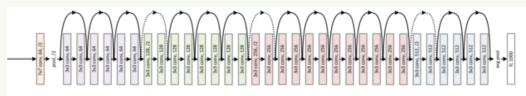
$x_{n+1} = x_n + hf(t_n + x_n)$

Residual Neural Network (ResNet)

$y_n = x_n + F(x_n); \quad x_{n+1} = I(y_n)$

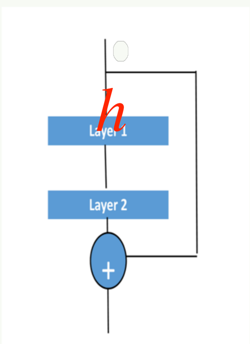






Euler-viewed ResNet

$y_n = x_n + h F(x_n); \quad x_{n+1} = I(y_n)$

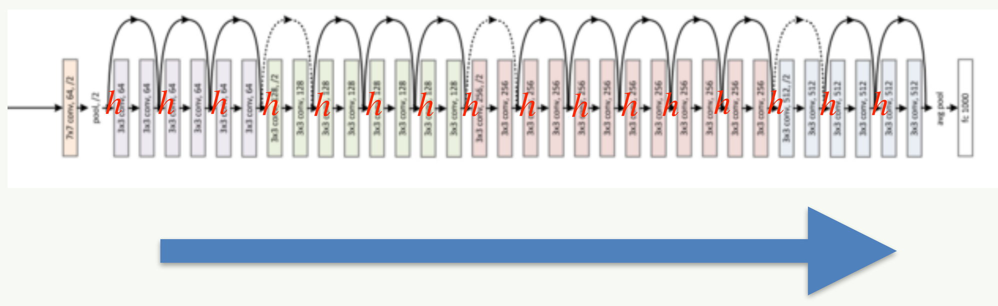


How step factor h helps ResNet?

Our Contribution:
We introduce a hyperparameter step factor h into Residual Neural Network (ResNet), theoretically and empirically confirming its efficacy in ResNet in term of **training robustness** and **generalization robustness**.

Training robustness refers to the stability of model training with an increasing depth, a larger learning rate, and different types of optimizer.
Generalization robustness refers to how well a trained model generalizes to classify test data whose distribution may not match that of the training data.

Small h on Generalization Robustness

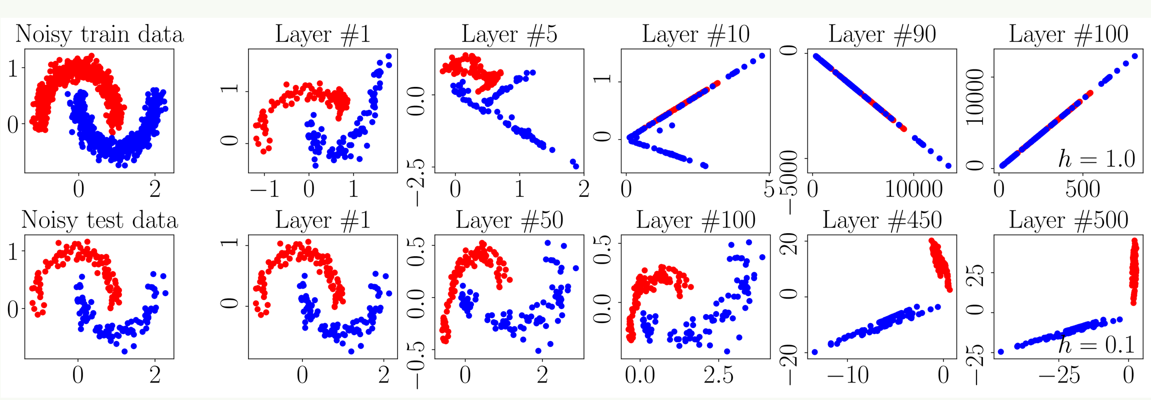


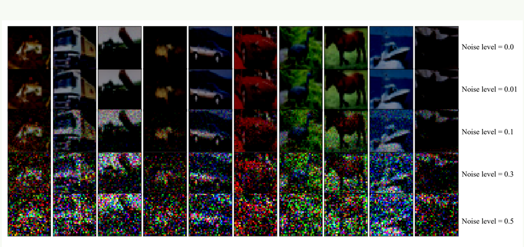
x_0 x_D

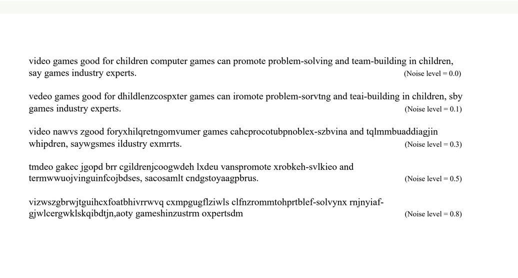
x_0^ϵ x_D^ϵ

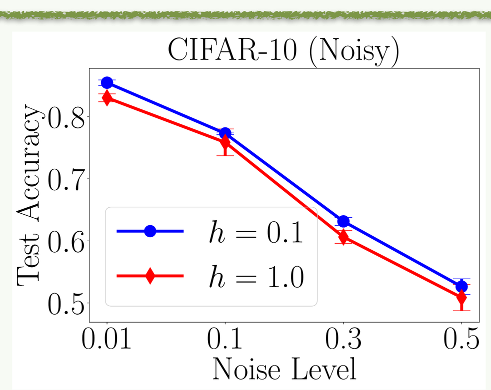
$$\epsilon_D = \|x_D^\epsilon - x_D\| \le \epsilon + hDW$$

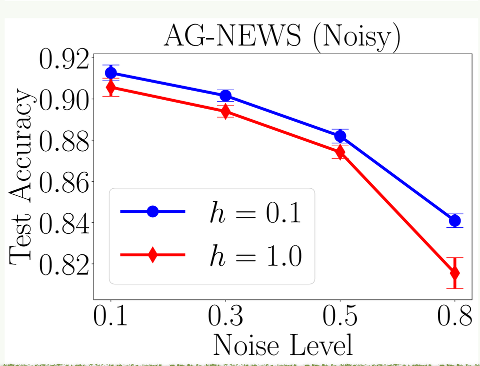
Small h can limit the noise amplification and provide the extra capacity for filtering out the input noise.







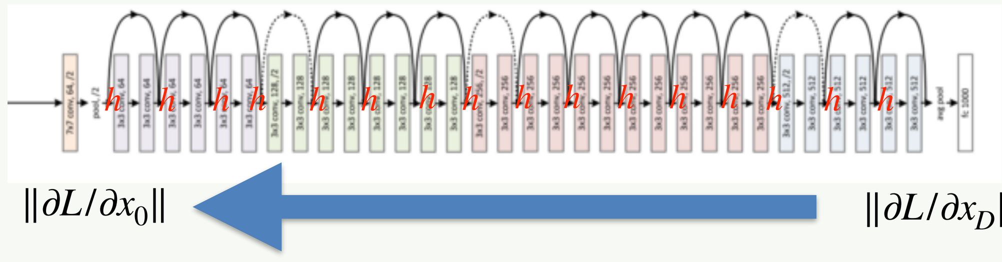




To verify the effectiveness of small h on generalization robustness of ResNet, we train on noisy input data with various noise level, and compare ResNet with reduced step factor (h = 0.1) and original ResNet (h = 1) on the clean test data.

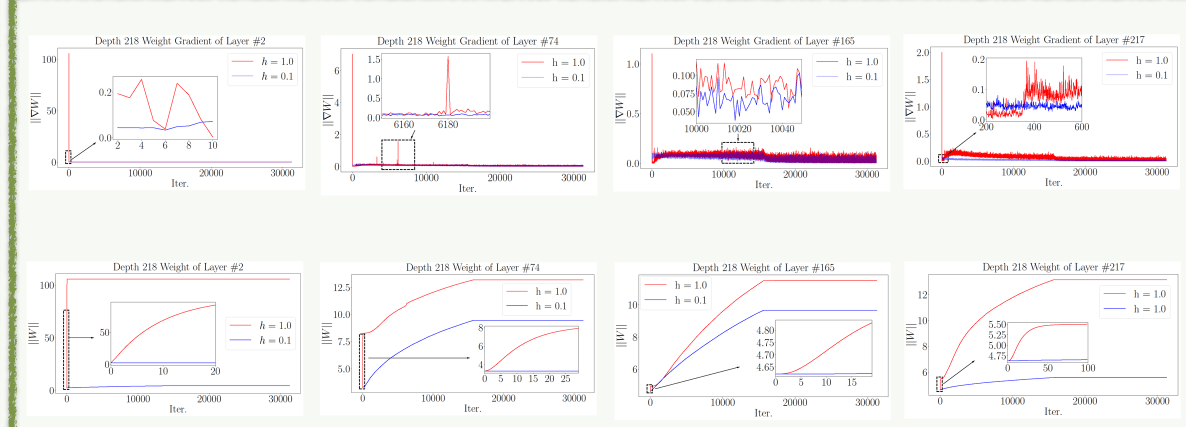
*Small h can ameliorate the input noise.

Small h on Training Robustness



$\|\partial L/\partial x_0\|$ $\|\partial L/\partial x_D\|$

$$\|\partial L/\partial x_0\| \le \|\partial L/\partial x_n\|(\sqrt{d} + hW)^D$$

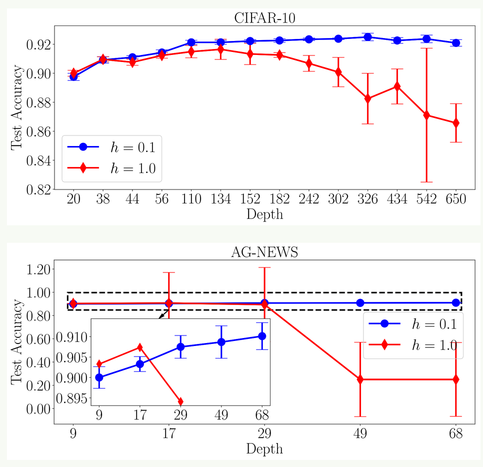


For existing method (h = 1.0, red line), ResNet has (a) exploding gradients problem (unstable training) and (b) large weight problem (input noise amplification over depth).

How the existing tricks handle above problems?
1 Warming up the training with a small LR
2 Gradient clipping
3 Weight decay (regularization)

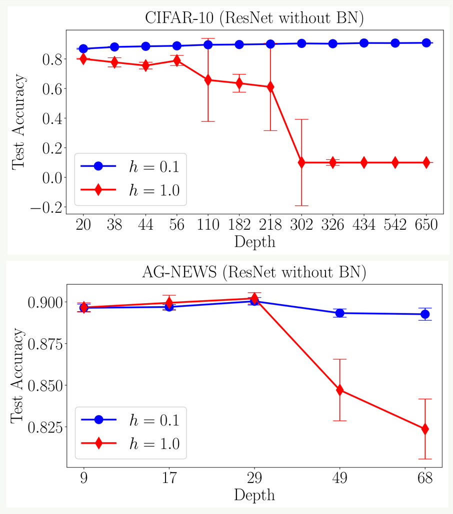
•By comparison, our method (h = 0.1) naturally has stable gradients and smaller weights!!!
•*Our method is compatible with existing techniques.

Small h helps training deeper ResNet



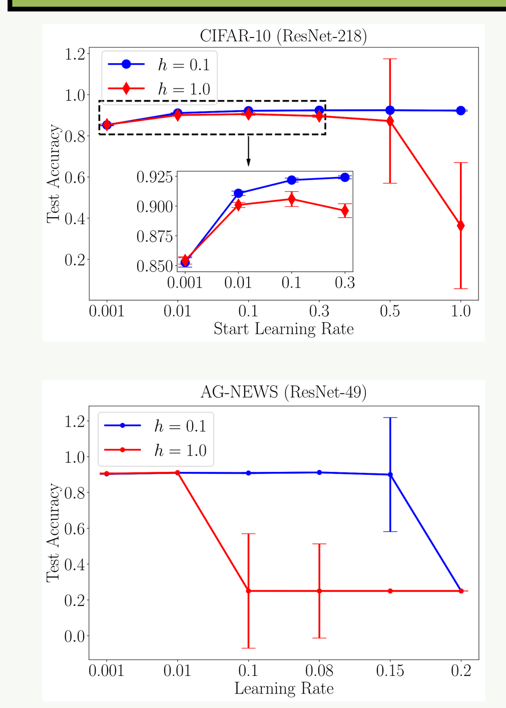
Our method (smaller h, .e.g. h = 0.1) makes the training procedure of ResNet more robust to increasing depth.

Small h helps without applying Batch Normalization



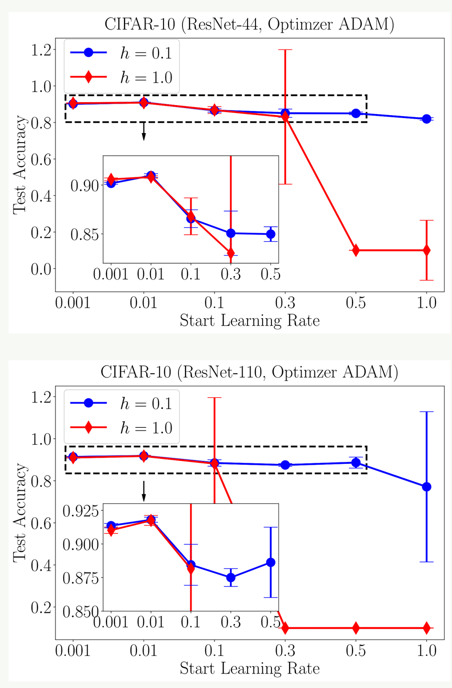
BN can help to stabilize the training procedure of DNN. Without applying BN, our small h can still stabilize the training by preventing the explosion of back-propagated information.
*Our small h is compatible with BN.

Small h can enable larger learning rate (SGD with momentum)



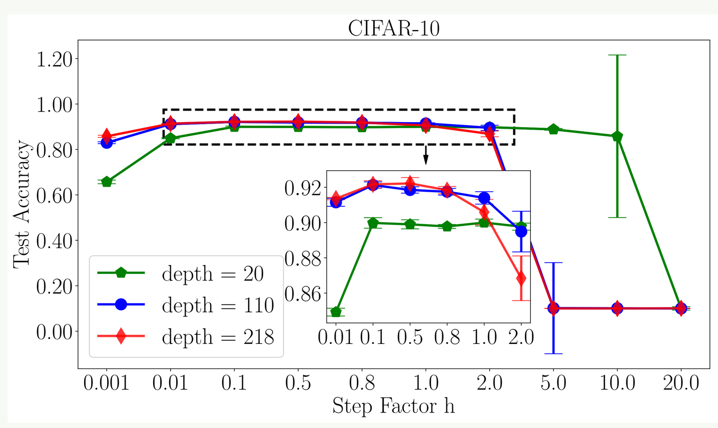
Training with smaller h (h = 0.1) is robust to larger LR with little chance of performance degradation.
For h = 1.0, gradient is unstable and bumpy over training iterations so that it requires careful adaptation of LR. By contrast, for h =0.1, gradient are small and stable, which enable larger LR.

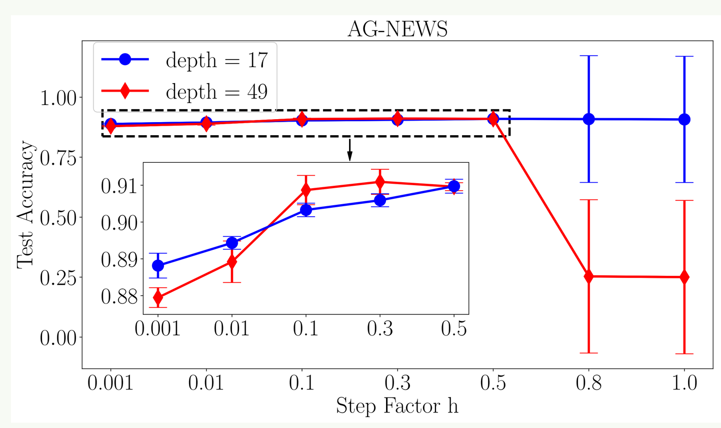
Small h even works on different optimizer - ADAM



We train ResNet using another popular optimizer - ADAM. Our small h still exhibits the competitive training robustness.

How to Select Step Factor h





Too large h will lead to unstable training.
Too small h will smooth out the useful transformations.

The take-away guidance is "smaller but not too small".

Acknowledgements

This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Strategic Capability Research Centres Funding Initiative, RIKEN-AIP, and IBM Singapore.