SAND2022-12733C

Examining stiffness in ResNets through interpretation as discretized Neural ODEs.

Joshua Hudson (PI) Khachik Sargsyan Marta D'Elia Habib Najm

LDRD Project #21-0528 FY21 – FY24

Sandia National Laboratories

¶ Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia LLC, a wholly owned subsidiary of Honeywell International Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

September 19, 2022

Introduction to Neural ODEs

Neural Ordinary Differential Equations (Neural ODEs)

ResNet

Indexed by layer: $n = 1, \ldots, N$

$$x_n = x_{n-1} + \alpha_n F_n(x_{n-1})$$

$$F_n(x) = \sigma\left(W_n x + b_n\right)$$

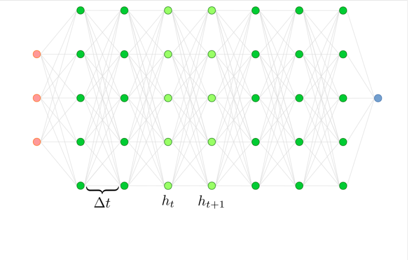
Neural ODE

Indexed by time: $t \in [0, T]$

$$\dot{x}(t) = F(x(t), t), \quad x(0) = x_0$$

$$F(x, t) = \sigma\left(W(t)x + b(t)\right)$$

- As $N \to \infty$, ResNet $\to$ a Neural ODE.
  - Well-posedness of the Neural ODE depends on smooth parameterization (interpolation) of weights and biases.
  - Scaling ResNet layer updates with $\alpha_n = T/N$ introduces the time scale: $\Delta t = T/N, \quad t = (T/N)n$.
  - **Infinite depth** interpretation
- Neural ODE discretized with explicit Euler scheme gives a ResNet.
  - Fix $N > 0$, $t_n := \frac{T}{N} n$, $W_n = W(t_n)$, $b_n = b(t_n)$:
  - Output $x_N \approx x(T)$



Depiction of ResNet convergence to NODE

Neural Ordinary Differential Equations (Neural ODEs)

ResNet

Indexed by layer: $n = 1, \dots, N$

$$x_n = x_{n-1} + \alpha_n F_n(x_{n-1})$$

$$F_n(x) = \sigma(W_n x + b_n)$$

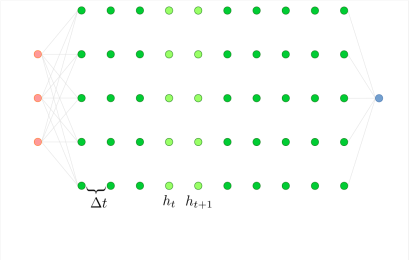
Neural ODE

Indexed by time: $t \in [0, T]$

$$\dot{x}(t) = F(x(t), t), \quad x(0) = x_0$$

$$F(x, t) = \sigma(W(t)x + b(t))$$

- As $N \to \infty$, ResNet $\to$ a Neural ODE.
  - Well-posedness of the Neural ODE depends on smooth parameterization (interpolation) of weights and biases.
  - Scaling ResNet layer updates with $\alpha_n = T/N$ introduces the time scale: $\Delta t = T/N, \quad t = (T/N)n$.
  - **Infinite depth** interpretation
- Neural ODE discretized with explicit Euler scheme gives a ResNet.
  - Fix $N > 0$, $t_n := \frac{T}{N} n$, $W_n = W(t_n)$, $b_n = b(t_n)$:
  - Output $x_N \approx x(T)$



Depiction of ResNet convergence to NODE

Neural Ordinary Differential Equations (Neural ODEs)

ResNet

Indexed by layer: $n = 1, \ldots, N$

$$x_n = x_{n-1} + \alpha_n F_n(x_{n-1})$$

$$F_n(x) = \sigma\left(W_n x + b_n\right)$$

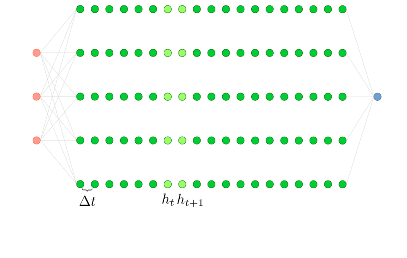
Neural ODE

Indexed by time: $t \in [0, T]$

$$\dot{x}(t) = F(x(t), t), \quad x(0) = x_0$$

$$F(x, t) = \sigma\left(W(t)x + b(t)\right)$$

- As $N \to \infty$, ResNet $\to$ a Neural ODE.
  - Well-posedness of the Neural ODE depends on smooth parameterization (interpolation) of weights and biases.
  - Scaling ResNet layer updates with $\alpha_n = T/N$ introduces the time scale: $\Delta t = T/N, \quad t = (T/N)n$.
  - **Infinite depth** interpretation
- Neural ODE discretized with explicit Euler scheme gives a ResNet.
  - Fix $N > 0$, $t_n := \frac{T}{N}n$, $W_n = W(t_n)$, $b_n = b(t_n)$:
  - Output $x_N \approx x(T)$



Depiction of ResNet convergence to NODE

Neural Ordinary Differential Equations (Neural ODEs)

ResNet

Indexed by layer: $n = 1, \ldots, N$

$$x_n = x_{n-1} + \alpha_n F_n(x_{n-1})$$

$$F_n(x) = \sigma\left(W_n x + b_n\right)$$

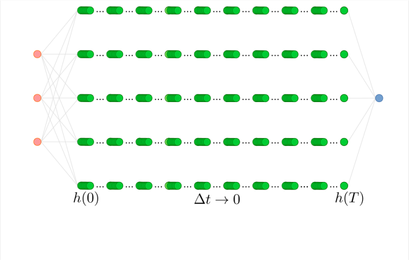
Neural ODE

Indexed by time: $t \in [0, T]$

$$\dot{x}(t) = F(x(t), t), \quad x(0) = x_0$$

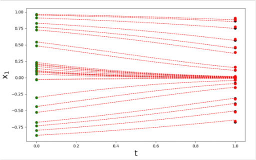
$$F(x, t) = \sigma\left(W(t)x + b(t)\right)$$

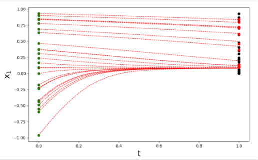
- As $N \to \infty$, ResNet $\to$ a Neural ODE.
  - Well-posedness of the Neural ODE depends on smooth parameterization (interpolation) of weights and biases.
  - Scaling ResNet layer updates with $\alpha_n = T/N$ introduces the time scale: $\Delta t = T/N, \quad t = (T/N)n$.
  - **Infinite depth** interpretation
- Neural ODE discretized with explicit Euler scheme gives a ResNet.
  - Fix $N > 0$, $t_n := \frac{T}{N}n$, $W_n = W(t_n)$, $b_n = b(t_n)$:
  - Output $x_N \approx x(T)$



Depiction of ResNet convergence to NODE

- Continuous vs discrete
  - Path crossing issue: a well-posed ODE has **backward uniqueness**
    ($x(t) = y(t) \implies x \equiv y$ on $[0, t]$.)
  - Can add an *extra dimension* to facilitate crossing (Dupont, *Augmented Neural ODEs*, NeurIPS 2019).





- Discretized comparison
  - Forward - Explicit Euler discretization of Neural ODE and ResNet are equivalent.
  - Backward - **gradients are different** due to differences in discretize-then-optimize and optimize-then-discretize approaches.

Linear layers with identical weights

Neural ODE: $\nabla loss = 2\left((1+\delta t W)^L x - y\right)(1+\delta t W)^L x$

ResNet: $\nabla loss = 2\left((1+\delta t W)^L x - y\right)(1+\delta t W)^{L-1} x$

Stiffness

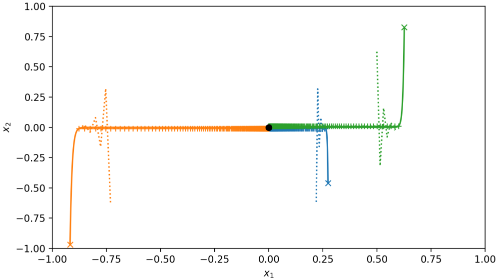
Stiffness in differential equations

Intuitive Idea of Stiffness

The existence of a large gap between the timescales at which coupled states evolve.

- Necessitates the continued use of a much smaller timescale (for stability purposes) to resolve the overall dynamics, even after the faster evolving processes have become exhausted.

- Linear system example:
  - $\dot{x} = Ax$
  - eigenvalues of $A = \lambda_1, \lambda_2, \ldots, \lambda_n$
    - $real(\lambda_1) < \ldots < real(\lambda_n) < 0$
  - Stiffness ratio: $r(A) = \frac{real(\lambda_1)}{real(\lambda_n)}$
  - Case depicted on the right: $A = \begin{bmatrix} -1 & 0 \\ 0 & -100 \end{bmatrix}$, $dt = 10^{-4}$ initial, $dt = 0.03$ after exhaustion of fast process ($|x_2| < 0.01$)



Stiffness in ResNets

- Literature:
  - Kim et al, *Stiff Neural Ordinary Differential Equations*, (2021) arXiv:2103.15341
  - Ghosh et al, *STEER: Simple Temporal Regularization For Neural ODEs*, (2020) arXiv:2006.10711
- Intuition from NODE: nth layer's rate of change is $\frac{x_n - x_{n-1}}{\alpha_n} = F_n(x_{n-1})$
  - Jacobian (linear part) of nth layer's rate of change:

$$J_n(x_{n-1}) := \nabla_{x_{n-1}} F_n(x_{n-1}) = \sigma'(W_n x_{n-1} + b_n) W_n$$

  - Compute stiffness of Jacobian for each layer n and sample i.
    - Stiffness of layer n for sample i: $s_{n,i} = r(\sigma'(W_n x_{n-1}^{(i)} + b_n) W_n)$
  - Sum stiffness across layers: total stiffness of the ResNet: $\sum_{i,n} s_{n,i}$
- Reducing stiffness to improve prediction performance
  - Penalizing Stiffness
    - Multiply sum by a weight (*Lagrange multiplier*) and add to MSE as total *loss* to be minimized.

Measuring Stiffness

- Directly compute eigenvalues of Jacobian (J): $\tilde{r}(J) = \frac{|\lambda_1|}{|\lambda_n|}$, $|\lambda_1| > ... > |\lambda_n|$
  - Analytic formulae for eigenvalues (dimension less than five).
  - General eigenvalue solvers **not easily differentiable!**
- Differentiable proxies for stiffness
  - Singular values (symmetric eigenvalue solvers are differentiable)
    - $\sigma_1 > |\lambda_1| > |\lambda_n| > \sigma_n$.
  - (Complex) power-method computes the spectral radius (i.e. $|\lambda_1|$).
    - Iteration: $v_{n+1} \leftarrow Jv_n$.
    - v_n tends to eigenspace of dominant eigenvalue (may not converge).
  - Gelfand's formula for the spectral radius: $|\lambda_1| \approx \|J^k\|^{\frac{1}{k}}$
    - *Implemented with $k = 2^{10}$ using a sequence of 10 squarings and* ***normalizations*** *for numeric stability.*

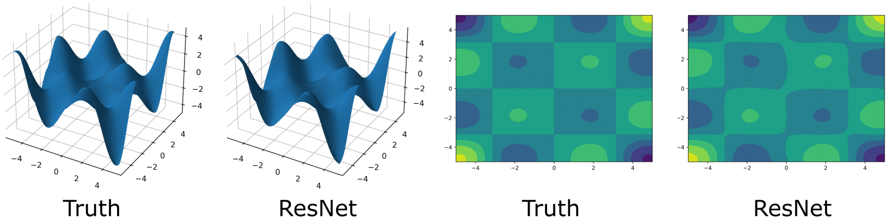
Test Case

Learning task

- Alpine 02 test case
- A benchmark problem to test optimization algorithms

$$f_{A2}(x) = \prod_{i=1}^{d} \sqrt{|x_i|} \sin(x_i).$$

- We will use it here as a regression test problem, where we try to learn the mapping $x \mapsto f_{A2}(x)$ for points $x \in [-5, 5]^d \in \mathbb{R}^d$.

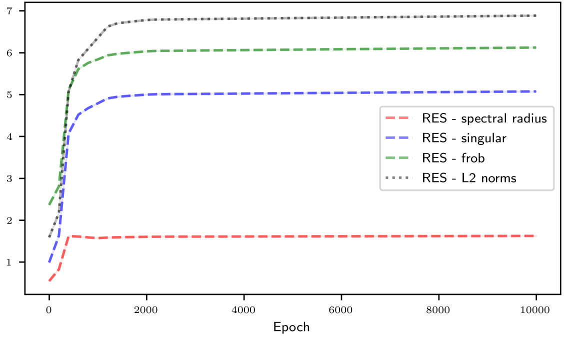


Visual comparison of ResNet approximation of f_{A2} after training.

Stiffness proxy evolution for test case

- ResNet architecture
  - Width: 20
  - Depth: 10
  - Activation: tanh
- Training
  - 900 training points
    - 20 mini-batches of 45 samples
  - 100 test points
  - 10k epochs
  - Optimizer: ADAM
    - Adaptive learning rate using pytorch's *reduce on plateau*.
    - Initial learning rate: 1.0e-3.

Model	ER	GE	df	frob	singular
RES	0.00389	0.00613	2.64	37.5	25.8

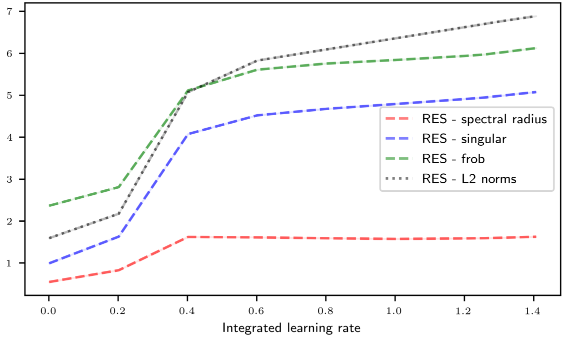


Evolution of stiffness proxies

Stiffness proxy evolution for test case

- ResNet architecture
  - Width: 20
  - Depth: 10
  - Activation: tanh
- Training
  - 900 training points
    - 20 mini-batches of 45 samples
  - 100 test points
  - 10k epochs
  - Optimizer: ADAM
    - Adaptive learning rate using pytorch's *reduce on plateau*.
    - Initial learning rate: 1.0e-3.

Model	ER	GE	df	frob	singular
RES	0.00389	0.00613	2.64	37.5	25.8



Evolution of stiffness proxies

Penalizing stiffness - numerical study

Climate model test case

Learning task

- Predict the output of a high-fidelity climate model.
  - Input dimension: 15
  - Output dimension: 10
  - Energy Exascale Earth System Model (E3SM) [Golaz 2022] Land Model (ELM) version 2
  - Vegetation dynamics resolved via the Functionally Assembled Terrestrial Ecosystem Simulator (FATES) [Koven 2020]

Models

- RES: ResNet trained without penalization
- RES L2: ResNet with L2 regularization
  - Penalize average of Euclidean/Frobenius norms of all network parameters.
  - Penalty weight: $\lambda = 10^{-5}$
- RES stiff: ResNet trained with stiffness penalization
  - Spectral radius used as the stiffness proxy, computed using Gelfand's formula.
  - Penalty weight: $\lambda = 10^{-3}$

ResNet architecture

- Width: 50
- Depth: 16
- Activation: tanh

Training

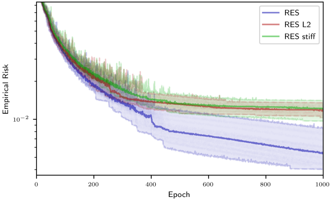
- 1996 training points
  - 100 mini-batches of 20 samples
- 500 test points
- 1k epochs
- Optimizer: ADAM
  - Adaptive learning rate using pytorch's *reduce on plateau*.
  - Initial learning rate: 1.0e-3.
- Loss: quadratic mean of RMSE and penalty:

$$\sqrt{\sum_{(x_i, y_i) \in S} |F(x_i) - y_i|^2 + \lambda p_i^2}$$

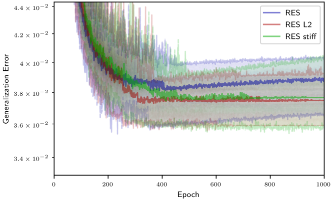
Penalization Reduces Generalization Error in Climate Test Case I

Summary of results for 25 trials (initializations)

Model	penalty	ER	GE	stiffness	wall time (sec)
RES (med)	0.00	5.4e-03	3.9e-02	3.52	17903
RES L2 (med)	25.76	1.2e-02	3.7e-02	2.61	23547
RES stiff (med)	0.05	1.2e-02	3.8e-02	0.05	1848662

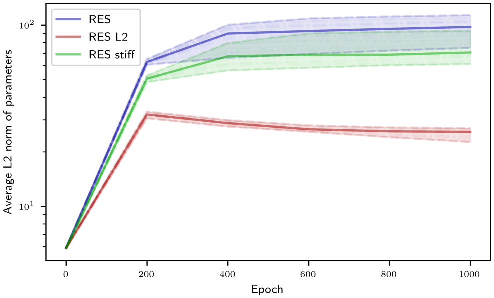


Evolution of training error (empirical risk)

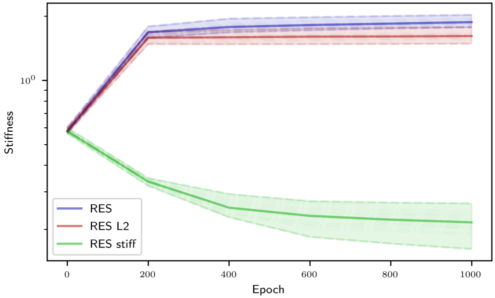


Evolution of testing error (generalization error)

Penalization Reduces Generalization Error in Climate Test Case II



Evolution of L2 norms



Evolution of stiffness

Questions?