Advanced Neural Ordinary Differential Equation Solver Using PETSc

Wenjun Zhao
Givens Associate
Argonne National Laboratory
Ph.D. Candidate at New York University

Mentor: Dr. Hong Zhang
Assistant Computational Mathematician
Argonne National Laboratory

August 21st, 2020
New York City, NY

Introduction to Neural ODEs

- Residual networks: with f as a neural network,

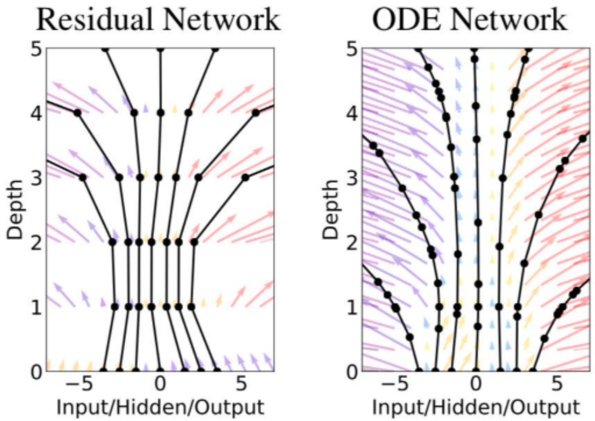
$$z_{t+1} = z_t + f(z_t, \theta_t), \qquad t \in \{0,1,\dots,T\},$$

which can be interpreted as an Euler discretization of an ODE.

- **Neural ODE[1]:** in the limit of smaller time steps,

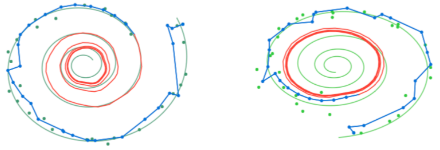
$$\frac{dz}{dt} = f(z(t), \theta(t), t), \qquad z(0) = z_0,$$

and the output is defined by $z(T)$.

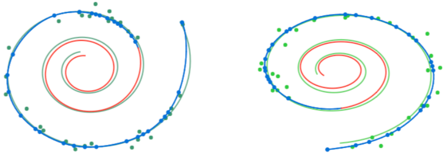


Applications:

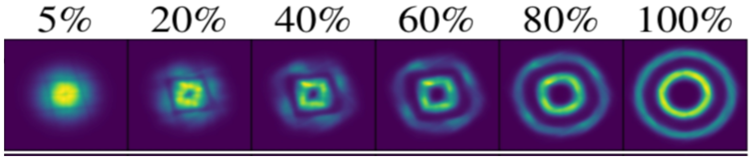
- Replacing residual blocks in supervised learning[1] ;
- Time series modeling[1] ;
- Density estimation through continuous normalizing flows[1][2], …,



(a) Recurrent Neural Network



(b) Latent Neural Ordinary Differential Equation



Different implementations

To calculate the gradient for the loss function, the adjoint method is applied:

Forward: $z(t_1) = z(t_0) + \int_{t_0}^{t_1} f(z,t)dt$

Backward: $a(t_0) = a(t_1) + \int_{t_1}^{t_0} a(t)^T f_z(z,t)dt$, where $a(t) = L_z(z(t))$ is the adjoint state

Gradient: $L_\theta = \int_{t_0}^{t_1} a(t)^T f_\theta(z,t)dt$

Our approach:

We use discrete adjoint method and checkpointing strategy through PETSc TSAdjoint **[6]**

We compare it with the following 3 existed implementations:

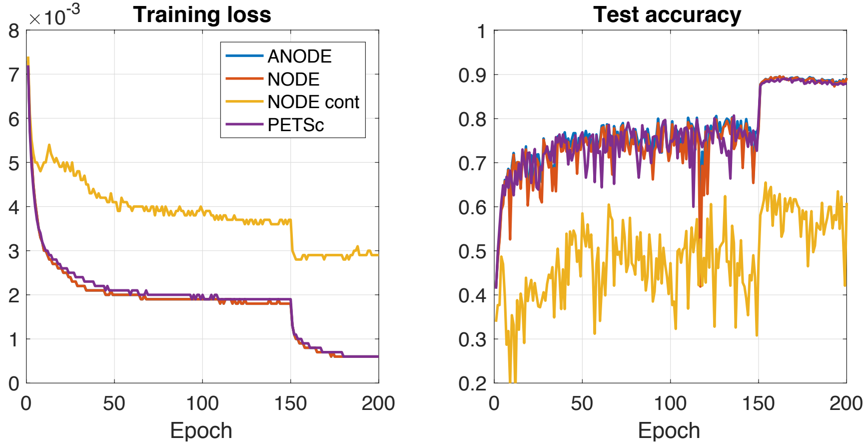
- Discrete adjoint, checkpointing everything (referred as NODE [1])
- Continuous adjoint, no checkpointing (referred as NODE cont.[1])
- Discrete adjoint, multi-level checkpointing (referred as ANODE [3])

Accuracy

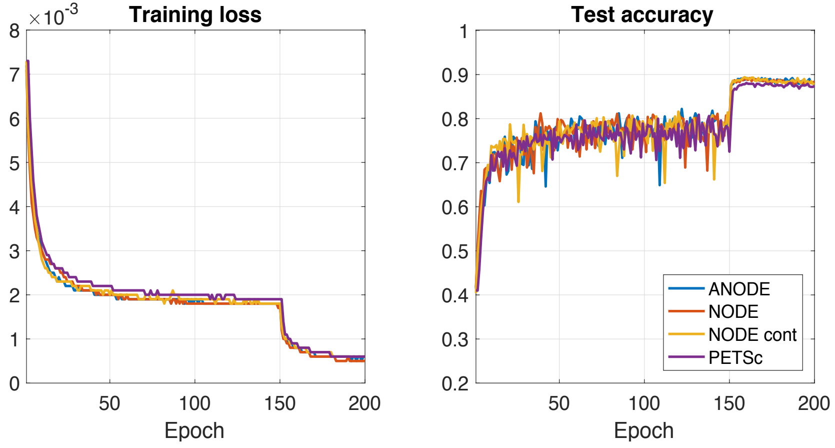
- NODE, ANODE, PETSc use **discrete adjoint** method, which provides exact gradients;
- NODE cont uses **continuous adjoint**, has large error with low accuracy time steppers.

Example: Classification on Cifar-10 dataset, architecture with 4 ODE blocks [3]

(1) With forward Euler method (first order):



(2) With Runge-Kutta 4 method (4-th order):



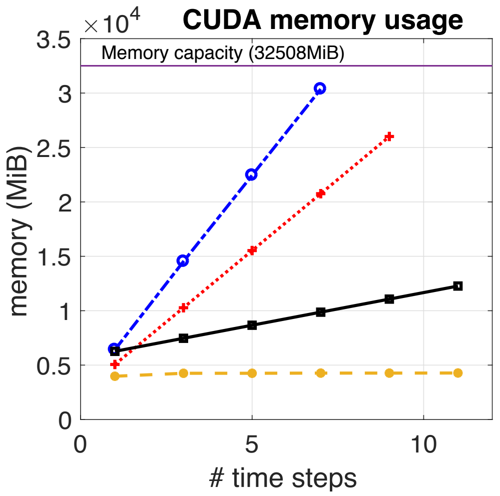
Efficiency

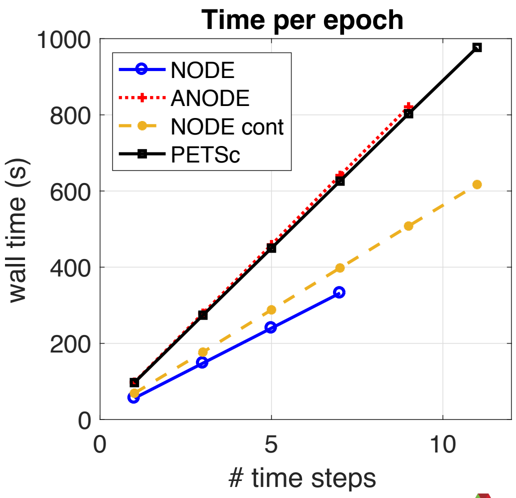
- NODE **checkpoints everything** (function evaluations and trajectory) in the forward pass;
- NODE cont. reverses the ODE and **does not require checkpointing**;
- ANODE **saves only initial conditions** and recomputes from the checkpoints;
- PETSc implementation **saves the complete trajectory** but not function evaluations;

L: # ODE blocks; Nt: # time steps
NFE: # function evaluations

	Mem	Time	NFE
NODE	O(LNt)	O(LNt)	O(LNt)
NODE cont.	O(1)	O(2LNt)	O(2LNt)
ANODE	O(L+Nt)	O(2LNt)	O(2LNt)
PETSc	O(LNt)	O(2LNt)	O(2LNt)

Example: Classification on Cifar-10 dataset, with 4 ODE blocks, solved through Runge-Kutta 4



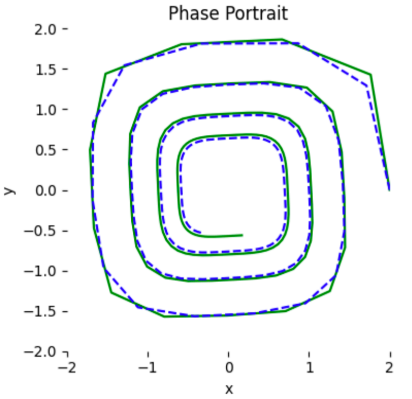


Implicit methods for neural ODEs

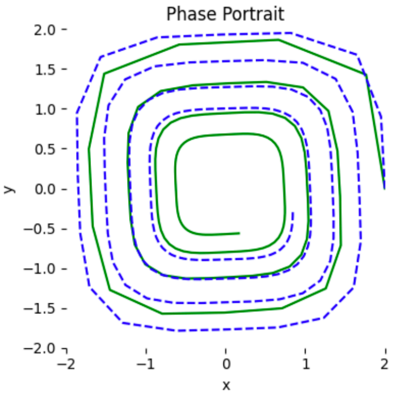
- PETSc offers sophisticated linear and nonlinear solvers that allows for implicit time stepping;
- Implicit methods are known for **absolute stability with arbitrary step size**, always find non-blow-up solutions during training and test.

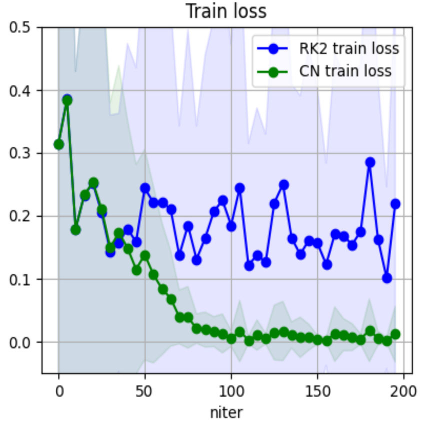
Example: a 2D toy model

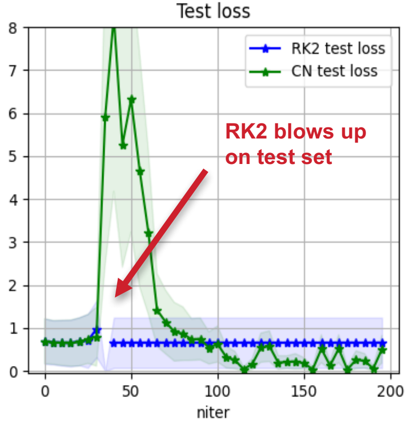
(1) With Crank-Nicolson (2nd order, implicit)



(2) With Runge-Kutta 2 (2nd order, explicit)







References:

[1] *Neural Ordinary Differential Equations,* Ricky T. Q. Chen, Yulia Rubanova, Iesse Bettencourt, David Duvenaud, Advances in Neural Information Processing Systems (NeurIPS), 2019

[2] *FFJORD: Free-form Continuous Dynamics For Scalable Reversible Generative Models,* Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, David Duvenaud, International Conference On Learning Representations (ICLR), 2019

[3] *ANODE: Unconditionally Accurate Memory-efficient Gradients For Neural Odes,* Amir Gholami, Kurt Keutzer, George Biros, International Joint Conferences On Artificial Intelligence (IJCAI'19), 2019.

[4] *Discretize-optimize Vs. Optimize-discretize For Time-series Regression And Continuous Normalizing Flows,* Derek Onken, Lars Ruthotto, arXiv:2005.13420

[5] *PETSc/TS: A Modern Scalable ODE/DAE Solver Library,* Shrirang Abhyankar, Jed Brown, Emil M. Constantinescu, Debojyoti Ghosh, Barry F. Smith, Hong Zhang, arXiv:1806.01437

[6] *PETSc TSAdjoint: A Discrete Adjoint ODE Solver For First-order And Second-order Sensitivity Analysis*, Hong Zhang, Emil M. Constantinescu, Barry F. Smith, arXiv: 1912.07696

Thank you!