An Empirical Study of Neural Ordinary Differential Equations

Linjian Ma, Yufan Sun, Chiyu Ding
{linjian, yufan_sun, chiyuding}@berkeley.edu
CS282 FANFANZUIBANG

Project Overview

We empirically study the new family of deep neural network models, neural ODE, which parameterize the derivative of the hidden state using a neural network, instead of specifying a discrete sequence of hidden layers. We experimentally verified the memory and model efficiency of neural ODE compared with the traditional ResNet and GRU model in image classification and time series prediction tasks, respectively. In addition, we demonstrate the ability of neural ODE in continuous function fitting.

Problem Statement

The method of re-frame a neural network as an "Ordinary Differential Equation" enables people to use existent ODE solvers. Although the ODE network method is new, it has already been a breakthrough in AI field and has great potentials. So our team dived deep into the effect of ODE solver on neural network training through empirical studies. In this project, we attempt to answer three questions:

1. For classification applications, how does it compare to the efficient ResNet for well-known datasets (e.g. Cifar10)?
2. It has been shown that ODENet performs better than RNN on synthetic time-series data. Can it also perform well on application datasets?
3. Can Neural ODE be used to recover the original function based on the sampled data?

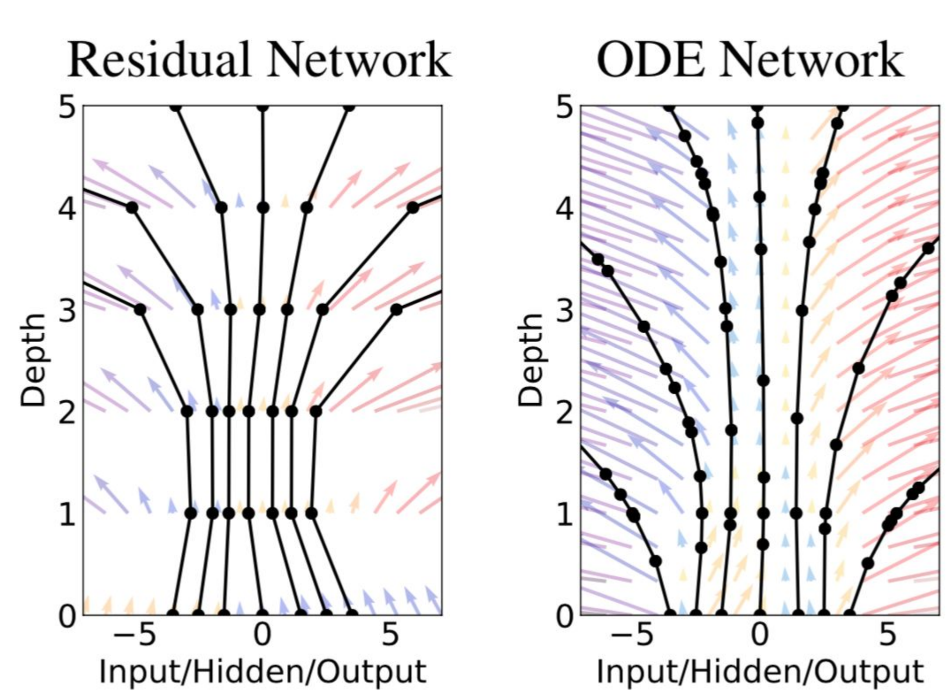
Neural Ordinary Differential Equations

For residual networks, the transformations within hidden units can be expressed by:

$$h_{t+1} = h_t + f(h_t, \theta_t)$$

For ODE method, the formulation above can also be regarded as an Euler discretization of a differential equation:

$$\frac{dh(t)}{dt} = f(h(t), t, \theta)$$



The equivalent of having T layers in ResNet, is finding the solution to ODENet at time 1.

Training:
An ODENet can be trained with the **Adjoint Method**, which can be regarded as the instantaneous analog of chain rule. This approach computes gradients by solving a second, augmented ODE backwards in time, and is applicable to all ODE solvers.

Algorithm 1 Reverse-mode derivative of an ODE initial value problem

Input: dynamics parameters θ, start time t_0, stop time t_1, final state $\mathbf{z}(t_1)$, loss gradient $\partial L / \partial \mathbf{z}(t_1)$

$s_0 = [\mathbf{z}(t_1), \frac{\partial L}{\partial \mathbf{z}(t_1)}, \mathbf{0}_{|\theta|}]$ ▷ Define initial augmented state

def aug_dynamics($[\mathbf{z}(t), \mathbf{a}(t), \cdot], t, \theta$): ▷ Define dynamics on augmented state

  return $[f(\mathbf{z}(t), t, \theta), -\mathbf{a}(t)^{\mathsf{T}} \frac{\partial f}{\partial \mathbf{z}}, -\mathbf{a}(t)^{\mathsf{T}} \frac{\partial f}{\partial \theta}]$ ▷ Compute vector-Jacobian products

$[\mathbf{z}(t_0), \frac{\partial L}{\partial \mathbf{z}(t_0)}, \frac{\partial L}{\partial \theta}] = \text{ODESolve}(s_0, \text{aug\_dynamics}, t_1, t_0, \theta)$ ▷ Solve reverse-time ODE

return $\frac{\partial L}{\partial \mathbf{z}(t_0)}, \frac{\partial L}{\partial \theta}$ ▷ Return gradients

References

[1] Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, pages 6572–6583, 2018.

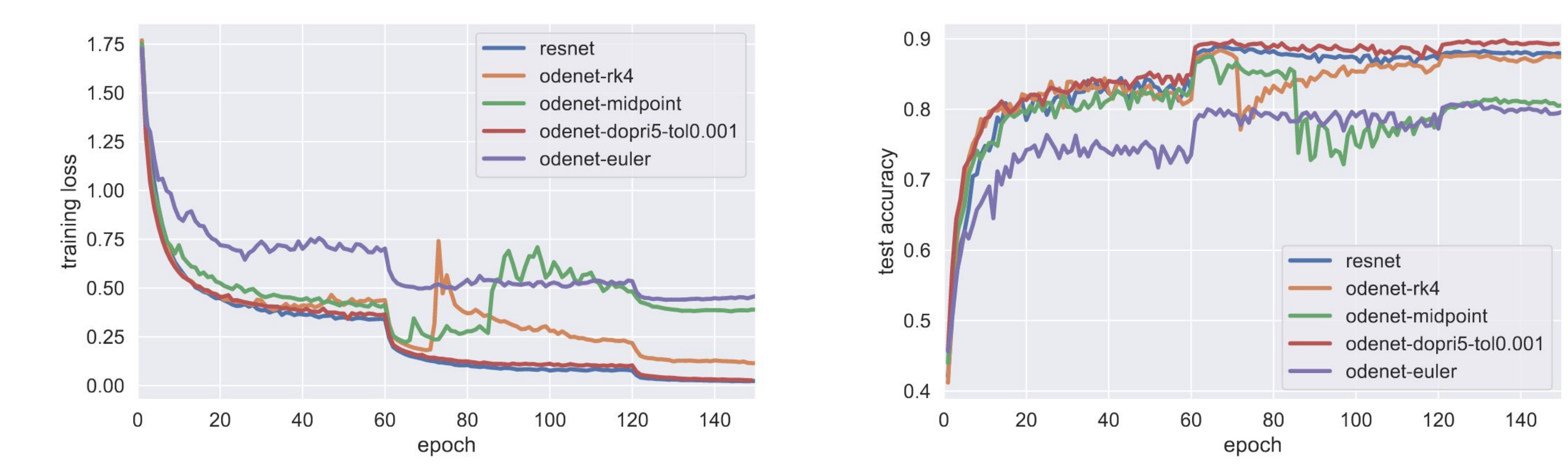
[2] Fox, Ian, et al. "Deep multi-output forecasting: Learning to accurately predict blood glucose trajectories." Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, 2018.

[3] Lev Semenovich Pontryagin.Mathematical theory of optimal processes. Routledge, 2018

Experiments and Results

1. Experiments on image classification

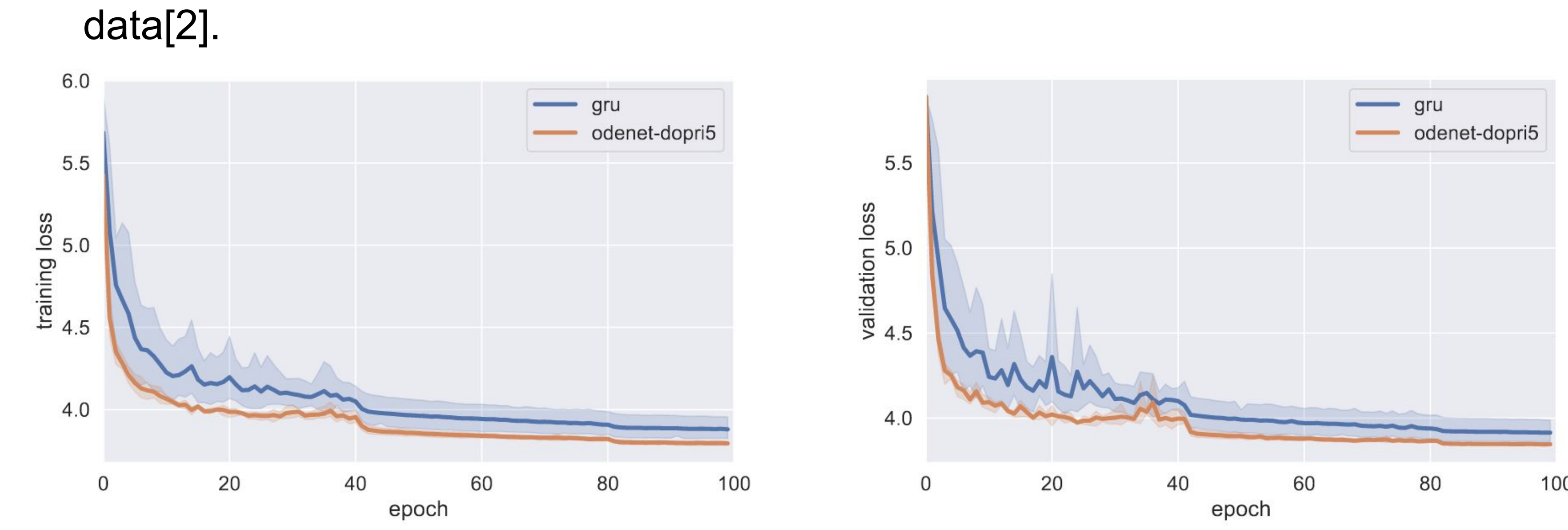
We tested Neural ODE on Cifar10. We show the training loss and test accuracy for various methods, and memory comparison for these ODE methods.



Update method	dopri5 (0.001 tolerance)	dopri5 (0.1 tolerance)	rk4	midpoint	euler
GPU memory	3060MB	2534MB	1701MB	1598MB	1367MB
forward iterations	21	17	4	2	1
backward iterations	24	15	5	3	2

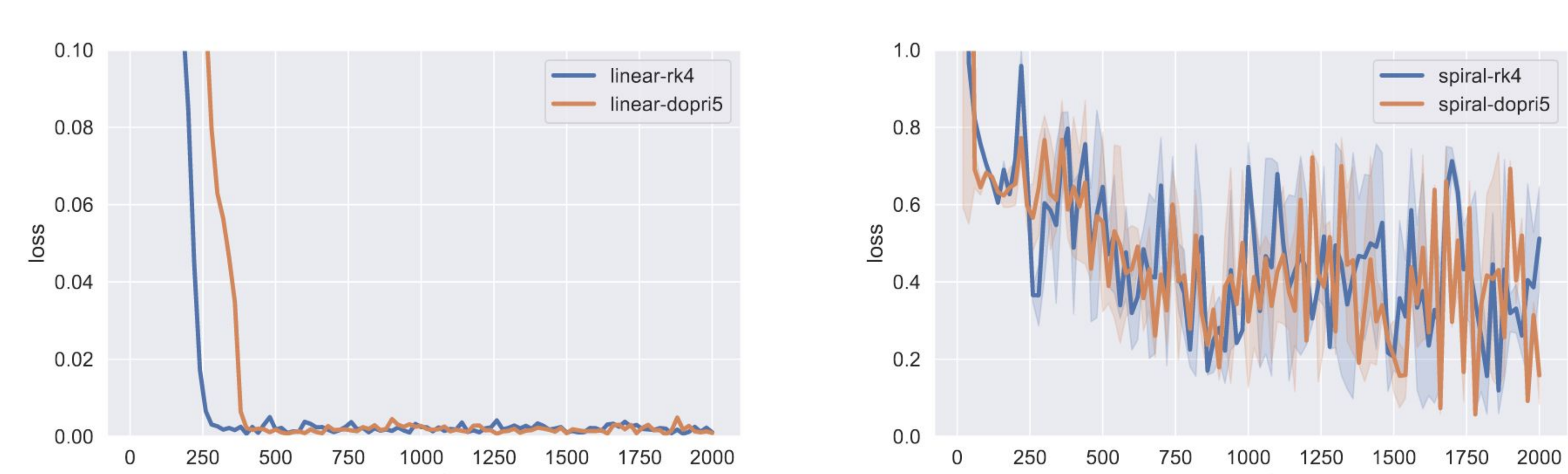
2. Experiments on latent time series prediction

We tested Neural ODE on predicting blood glucose trajectories time series data[2].



3. Experiments on function fitting

We tested the ability of Neural ODE to learn the true dynamics of some specific differential equations given sampled data. Left: Linear ODE, Right: Spiral shaped nonlinear ODE.



Results Discussion

Accuracy

Neural ODE can achieve the similar accuracy as, and sometimes better accuracy than ResNet and GRU based network.

Memory

Theoretically, less memory usage. Not storing any intermediate quantities of the forward pass allows us to train our models with constant memory cost as a function of depth.

Stability

Training with neural ODE with adjoint method is not that stable, which can be seen in our results. It it because numerical ODE algorithm is non-revertible. Using higher order methods (dopri5, rk4) can increase the stability.