Neural Ordinary Differential Equations

Pramook Khungurn

April 23, 2022

This is a note on the paper "Neural Ordinary Differential Equations" by Chen et al.[CRBD18].

1 Introduction

- Many existing neural networks models creates a sequence of hidden states $\mathbf{h}_0$, $\mathbf{h}_1$, $\mathbf{h}_2$, ... $\mathbf{h}_T$ by adding something to the previous state:

$$\mathbf{h}_{t+1} = \mathbf{h}_t + f(\mathbf{h}_t, t, \boldsymbol{\theta})$$

  Such models include such as residual networks [HZRS15], recurrent neural networks, and normalizing flows [RM15, DKB14].

- What if we take the limit as the number of time step goes to infinity? We will have a differential equation:

$$\frac{\mathrm{d}\mathbf{h}(t))}{\mathrm{d}t} = f(\mathbf{h}(t), t, \boldsymbol{\theta}).$$

- To use the network, we simply say that $\mathbf{h}(0)$ is the input layer, and the output is $\mathbf{h}(T)$ at some time T. The output can be found by solving the initial value problem, and this can be done by any black-box differential equation solver.

2 How to train a neural ODE model

- The problem with the above approach is that it is unclear how to train such a neural ODE model.
  - The computation of the solution can require a lot of time steps. Differentiating through these time steps to compute the gradient would requires saving a lot of information in memory.
- The good news is that there is a method to compute the gradient using constant memory (i.e., does not depend on the number of time steps). This is called the **adjoint sensitivity method**. It requires, however, an ODE solve, which can be done, again, by any ODE solver.
- To derive the method, we first change the notations.
  - The hidden state at time t is now denoted by $\mathbf{z}(t)$.
  - The start and end time is now t_0 and t_1 instead of 0 and T, respectively.

References

[CRBD18] Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[DKB14] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation, 2014.

[HZRS15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[RM15] Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows, 2015.