Training stiff neural ordinary differential equations in data-driven wastewater process modelling

https://doi.org/10.1016/j.compchemeng.2005.11.008

A pre-training approach to mitigate hardness in training stiff Neural ODE

Task

- System of ODE
  - given time derivative $\frac{d\boldsymbol{C}_t}{dt}$ and initial state $\boldsymbol{C}_0$, $\boldsymbol{C}_t$ can be numerically solved.
  - Forward: governing equation -> system states.
- Inverse problem
  - Given a series of observation of system states, we now want to estimate (parameters in) the governing equation.
  - Backward: system states -> governing equation.

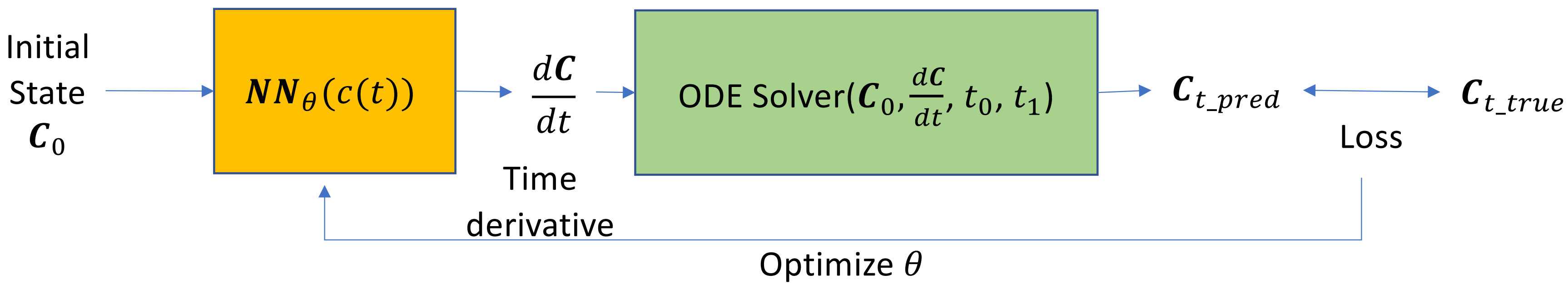
Neural ODE

Let's say that the following are all our included ozone related reactions:

1) $O_3 \rightarrow O^1D$, J1
2) $O \rightarrow O_3$, k2
3) $O + O_3 \rightarrow$ dummy, k3
4) $NO + O_3 \rightarrow NO_2$, k4
5) $NO2 + O_3 \rightarrow NO_3$, k5

$d[O_3]/dt = -J1*[O_3] + k2*[O] - k3*[O_3]*[O] - k4*[NO]*[O_3] - k5*[NO_2]*[O_3]$

- Modeling a system of ODE.
  - $\frac{d\boldsymbol{C}_t}{dt}$ = f($\boldsymbol{C}_t$, t)
- In chemistry kinetic, f() is a polynomial of $\boldsymbol{C}_t$ with unknow coefficient.
- NODE use a neural network to play the role of differential equation, by optimize the parameters in the neural network and minimize loss, we say the neural network approximates the governing equation.

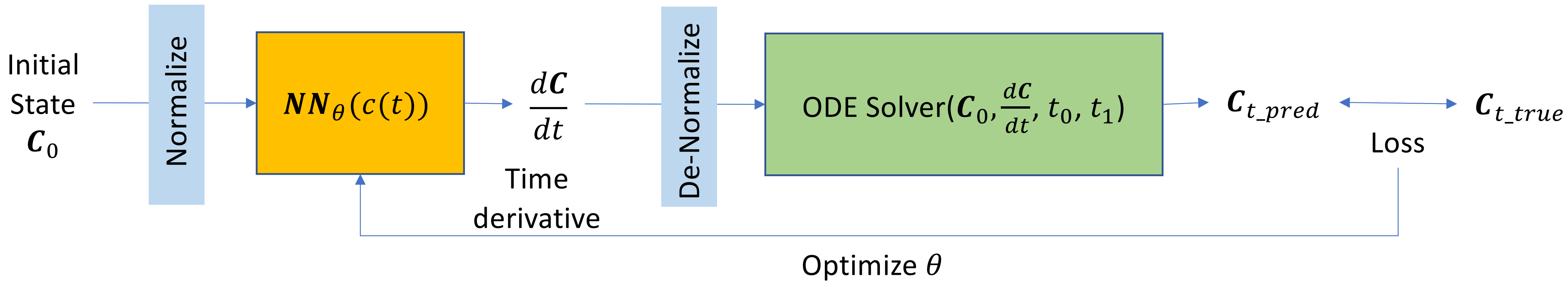


Hardness in stiff NODE

- Stiff ODE make system states varys at huge different speed => solving stiff ODE need very small time step.
- Error comes from:
  - Random weight initialization, gradient decent optimization introduce variation and randomness to Jacobian of the ODE. Aamplify errors, diverge training.
  - Gradient are large for fast variable and small for slow variable.

1. Normalization

- Normalize both state C(t) and de-normalize time derivative dC(t)/dt
  - De-normalize is important because it contians physical info
  - Normalize C(t) is simple, just min/max.
  - dC(t)/dt is unknow, so use ***difference quotients*** to estimate from C(t):
  - X’ = (x2 - x1, x3 - x2, ..., xn - xn-1)/Δt



2. Collocation Training

- We have C(t) and estimated dC(t)/dt, use this data to train a regression model first to interpolate data, and then pretrain the NODE model