Contributions

Black-box ODE solvers as a differentiable modeling component.
- Continuous-time recurrent neural nets and continuous-depth feedforward nets.
- Based computation with explicit control over tradeoff between speed and numerical precision.
- ODE-based change of variables for automatically-invertible normalizing flows.
- Open-sourced ODE solvers with O(1)-memory backprop:
  https://github.com/rqichen/torchdiffeq

ODE Solvers: How Do They Work?

- \( z(t) \) changes in time, defines an infinite set of trajectories.
- Define a differential equation: \( \frac{dz}{dt} = f(z(t), t, \theta) \).
- Initial-value problem: given \( z(t_0) \), find \( z(t) = z(t_0) + \int_{t_0}^{t} f(z(t), t, \theta) \, dt \).
- Approximate solution with discrete steps, e.g. \( z(t + h) = z(t) + hf(z, t) \).
- Higher-order solvers are more accurate and use larger step sizes.
- Can adapt step size \( h \) given error tolerance level.

Continuous version of ResNets

ODE-Nets replaces ResNet blocks with ODESolve\((f; z(t_0), t_0, t_1, \theta)\), where \( f \) is a neural net with parameters \( \theta \).

\[ z(t_1) = z(t_0) + \int_{t_0}^{t_1} f(z(t), t, \theta) \, dt = \text{ODESolve}(z(t_0), f, t_0, t_1, \theta) \]

ODE Nets for Supervised Learning

Adaptive computation: can adjust speed vs precision.
We can specify the error tolerance \( \theta \) of ODE Solve.

\[ \text{ODESolve}(f; z(t_0), t_0, t_1, \theta, rtol, atol) \]

Computing gradient for ODE solutions

- O(1) memory cost when training.
- Don’t store activations, follow dynamics in reverse.
- No backpropagation through the ODE solver – compute the gradient through another call to ODESolve.

\[ \text{def f_and_a}(x, z, t): \]
\[ \text{return } [z, +dt/dx, \theta, +dt/d\theta] \]

\[ \text{ODESolve}(f, z(t), x, \theta, 0, t_0, t_m) \]

Continuous Normalizing Flows (CNF)

Automatically-invertible Normalizing Flows.
Planar CNF is smooth and much easier to train than planar NF.

\[ \text{Planar normalization flow} \]
\[ \text{Continuous analog of planar flow} \]

Samples

Data

Figure: Visualizing the transformation from noise to data. Continuous normalizing flows are efficiently reversible, so we can train on a density estimation task and still be able to sample from the learned density efficiently.

Continuous-time Generative Model for Time Series

Time series with irregular observation times.
No discretization of the timeline is needed.

\[ \text{Figure: ODESolve} \]

Prior Works on ODE+DL

Chang et al. “Multi-level Residual Networks from Dynamical Systems View.” (2018)