DiffSim 24.02.

Ludwig Winkler

Machine Learning Group
TU Berlin

February 22, 2022

DiffSim - Idea

- Autodiff frameworks work great for machine learning: we compute gradients for parameters conveniently
- Why stop at computing parameter gradients? Why not autodiff arbitrary code?
- Big part of modern engineering/research are based on computer models/simulations

- Let's autodiff entire computer models/simulations!
- Question: What could we do with that?

What's happened so far

- Simulation of explicit high-dim simulation vs low-dim latent simulation [Rotating Gaussian distribution]
- Parts of latent simulation is done by neural network ODE

- Chaotic system with external force field [Double Pendulum]

- Three Body Problem of rotating planets [N-Body Problem]

What's to discuss

Problem of the day ... and months :(

- Ultimately we're training on a regression problem
- Long integration horizons aggravate basically tiny error
- Neural networks hide a lot of possibly random behaviour in their weights in unknown data regions

- Personal opinion: NN's bad for long term simulations (if we have the precise simulation)
- Classification doesn't care if it's 0.9 or 0.901, as it only wants the maximum

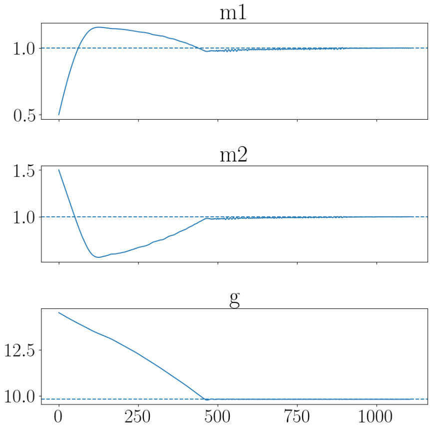
What's to discuss

Pivot Idea (and maybe what you wanted all along)

- Brenner AutoDiff Optimal Control paper
  - ML control on 2D Ising lattice model
  - ML control interacts in discreet way with simulation
  - ML control basically classification not regression

- Simulations offer precise computations (so let them do the heavy work)
- Autodiff through simulations to train hyperparameters of simulation

- Masses of joints (m1, m2) and gravitational constant g learned via DiffSim for Double Pendulum



What's to discuss

Pivot Idea (and maybe what you wanted all along)

- ML models trained via Reverse Mode AutoDiff
- Forward Mode AutoDiff allows differentiation of arbitrarily long simulations
- Simulation classes that could be tested out?

Next Step

- Require Forward Mode AutoDiff (ForAD) for simulations
- ForAD is memory independent
- Just keeps on multiplying memory constant Jacobians during simulation
- JAX and recently PyTorch offer ForAD

- Train on masses in planetary n-body problem via gradient training