Prof. Dr. Nils Thürey
Dr. Mario Lino, Benjamin Holzschuh, Patrick Schnell, Qiang Liu and Chengyun Wang
Technical University of Munich
School of Computation, Information and Technology
Summer Semester 2025

Advanced Deep Learning for Physics (IN2298)
Exercise 4
Supervised Network Training

In this worksheet, we consider a falling ball with negligible air friction. We will write a simple simulator that is used to train neural networks which should learn the forward dynamics.

(1) Analytic trajectories

Write a function that computes the exact state $(\vec{p}(t), \vec{v}(t))$ of a ball thrown at $t = 0$ from initial position $\vec{p}_0 = (x_0, y_0)$ with initial velocity $\vec{v}_0 = (v_x, v_y)$. The function arguments should be $(t, \vec{p}_0, \vec{v}_0)$. Assume gravity is the only force acting on the ball.

Then plot a couple of trajectories with various initial conditions to check that your implementation is correct.

(2) Simulation

Write a simulation function that takes a state and time increment $(\vec{p}, \vec{v}, \Delta t)$, simulates the evolution for Δt and returns the next state. Use Euler steps to integrate p and v (e.g., $p_{n+1} = p_n + \Delta t \cdot \frac{\partial p}{\partial t}$).

Then plot these trajectories next to the analytic trajectories from (a) for the same initial conditions. How do they differ? Show how the error grows with Δt.

(3) Data set

Generate a data set of initial conditions and corresponding trajectories, both analytic and simulated. Use a constant Δt to generate the trajectories. You don't need to store it on disk but it should be deterministic, i.e. identical every time you run your code.

(4) Neural corrector

Train a corrector neural network using the data set generated in part (c). The network should take a state computed by the simulator and learn a correction $\Delta\vec{p}, \Delta\vec{v}$ that can be added to the simulator output to get an approximation of the analytic solution.

Explain your loss function and analyze the network output. Simulate trajectories as in (a) and (b) using the simulator in combination with the corrector network and plot them along with the trajectories from (a) and (b). Can the network match the analytic solution exactly?

Hint if you are using PhiFlow

- You can set the random seed using `math.seed()`.
- You can use `dense_net()` to set up a simple fully-connected neural network. Also check out the neural network training example from the cookbook.

Submission instruction

Please upload a single PDF file containing your results along with your code for implementation tasks or your derivation for non-implementation tasks (LaTex typesetting). The uploaded PDF should only include the final code, so please trim empty spaces and your intermediate work before submitting.

The easiest way to generate such a PDF is by using Jupyter notebooks and LaTex (we recommend MiKTeX for Windows users). With Jupyter and LaTeX installed, you can create a PDF from your notebook by running *jupyter nbconvert –to pdf your-notebook.ipynb*

Additional information

This is an individual assignment. Plagiarism will result in the loss of eligibility for the bonus this semester.

If you have any questions about the exercises, please contact us via the forum on Moodle. If you need further face-to-face discussion, please join our weekly online Q&A session (every Monday at 15:00 and 16:00 via BBB).