Machine Learning Group
Faculty IV – Electrical Engineering and Computer Science
Technische Universität Berlin

Exercise Sheet 7

Exercise 1: Types of differential equations (10 P)

For a state $\underline{u} \in \mathbb{R}^2$, come up with an example for each of the following types of differential equations:

(a) linear, time-invariant ODE
An ODE that can be written in the form

$$\frac{\mathrm{d}\underline{u}(t)}{\mathrm{d}t} = \mathbf{A} \cdot \underline{u}\,, \quad \mathbf{A} \in \mathbb{R}^{2\times 2},$$

for example

$$\frac{\mathrm{d}\underline{u}(t)}{\mathrm{d}t} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot \underline{u} = \begin{bmatrix} u_1 + 2u_2 \\ 3u_1 + 4u_2 \end{bmatrix}.$$

(b) linear, time-variant ODE
An ODE that can be written in the form

$$\frac{\mathrm{d}\underline{u}(t)}{\mathrm{d}t} = \mathbf{A}(t) \cdot \underline{u}\,, \quad \mathbf{A}(t) \in \mathbb{R}^{2\times 2},$$

such that the parameters in $\mathbf{A}(t)$ change over time. For example

$$\frac{\mathrm{d}\underline{u}(t)}{\mathrm{d}t} = \begin{bmatrix} 1 & 2 \\ 3 & 4t \end{bmatrix} \cdot \underline{u} = \begin{bmatrix} u_1 + 2u_2 \\ 3u_1 + 4t \cdot u_2 \end{bmatrix}.$$

(c) non-linear, time-invariant ODE
An ODE that cannot be expressed as a linear combination of the states u_1 and u_2, for example

$$\frac{\mathrm{d}\underline{u}(t)}{\mathrm{d}t} = \begin{bmatrix} u_1^2 \\ 2u_2 \end{bmatrix}.$$

(d) non-linear, time-variant ODE
An ODE that cannot be expressed as a linear combination of the states u_1 and u_2 and where the parameters change over time. For example

$$\frac{\mathrm{d}\underline{u}(t)}{\mathrm{d}t} = \begin{bmatrix} u_1^2 \\ 2t \cdot u_2 \end{bmatrix}.$$

(e) partial differential equation
Example: transport equation

$$\frac{\partial \underline{u}(t,x)}{\partial t} + c\frac{\partial \underline{u}(t,x)}{\partial x} = 0$$

Exercise 2: Programming (70 P)

Download the programming files on ISIS and follow the instructions.

Exercise 3: Neural ODE classifier (20 P)

Using the neural ODE from the trajectory optimization problem in programming exercise 2, you now want to train a binary classifier on the "moons" dataset depicted in Figure 1a.

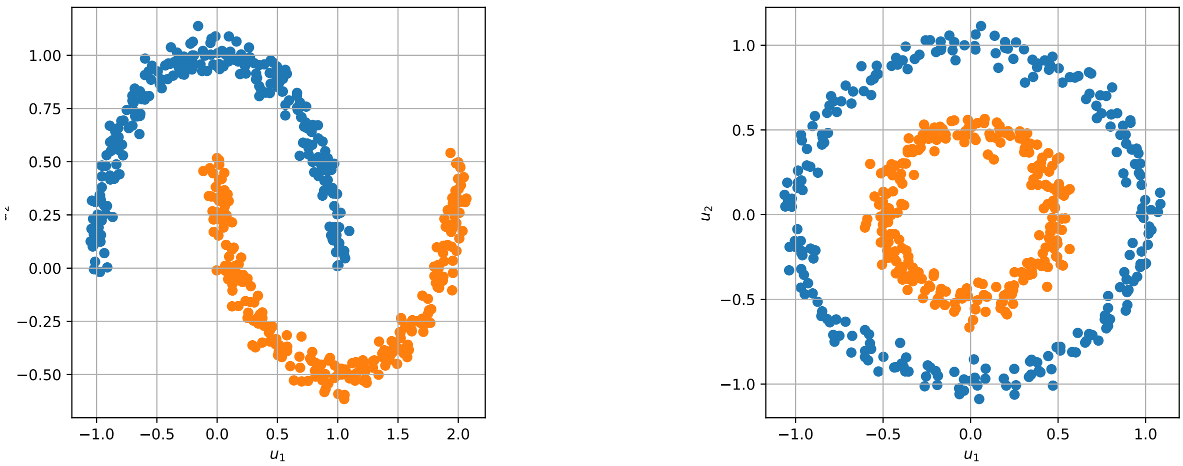
(a) In bullet-points, describe the changes you have to make to the existing code to train a classifier.

- define a data loader for the “moons” dataset
- integrate batches of data points
- use e.g. cross-entroy loss on terminal states $u_1(T)$ (or $u_2(T)$)

(b) In 2 to 3 sentences, explain why the “circles” dataset depicted in Figure 1b is difficult to classify for a Neural ODE and explain how this difficulty can be avoided.

The question refers to the figures in slide 20 on “Augmented Neural ODEs”:

- Neural ODE trajectories can’t intersect, they preserve the topology of the input space
- The neural ODE will learn a vector field that stretches one domain in the state space a lot
- This leads to high curvature in the trajectories and many small steps of the numerical integrator, which is computationally expensive
- Solution: use augmented Neural ODEs



(a) Moons

(b) Circles

Figure 1: Datasets for binary classification