Neural ODEs: breakdown of another deep learning breakthrough

Problem statement:

An Ordinary Differential Equation (ODE) describes how a variable, or a system of variables (space variables) behaves over time, according to one variable (time). As an example, in the real world, there is a model (a set of ODEs) in biology that describes the growth and interaction of populations over time, known as the Lotka-Volterra equations. However, modeling such equations analytically is not possible in all cases.

ODEs describe this behavior using derivatives, represented as

Since not all ODEs can be solved analytically, an alternative method for solving them is using numerical methods, such as the Euler method. The core idea behind numerical methods is to approximate the solution step by step using tangent lines. Therefore, if we know the right-hand side of the derivative, f(x, t), and the initial conditions (IC), we can approximate how the system changes over time after the given IC:

In real-life applications, this method allows us to make predictions for future steps over a considerable time interval. However, in the real world, finding the exact function of the system is difficult and often unknown. It is considerably easier to obtain data about the system for a certain time interval. If we somehow have access to we can learn the behavior of the system and predict future steps using numerical methods.

A Neural ODE is a neural network designed to mimic or approximate as a solution to this problem.

Data:

In a Neural ODE system, the data typically consists of input-output pairs. The input can take any form, whether structured or unstructured, such as images, time series, or text, depending on the specific application. For instance, if we are trying to model the population of foxes and rabbits in an ecosystem, the data would represent the population of each species over time. Additionally, in Neural ODEs, data is generated by a continuous-time process.

Model:

To approximate we create a feedforward neural network. The input and output size of the network should be equal to . This network is called the Neural ODE. The loss and parameter updates for the Neural ODE are similar to those of a regular neural network. The key difference is that we train the Neural ODE using a numerical method. We input the initial conditions, and the Neural ODE predicts derivative values for a given time interval. We then feed these derivative values into the numerical method to output system values for the same time interval and compare them with the true system values.

Link:

<https://towardsdatascience.com/neural-odes-breakdown-of-another-deep-learning-breakthrough-3e78c7213795>

<https://julialang.org/blog/2019/01/fluxdiffeq/>