

0.1 Nerual networks and PDE's

Neural networks can be useful for solving differential equations, . The Universal approximation theorem states that, under certain conditions, a neural network may approximate any well behaved function to any given precision. A differential equation describes a function by relating it to one of its own derivatives (to some order). In this way they do not describe the function explicitly, but rather its properties, and our job becomes looking for functions that exhibit these properties and thus are solutions to the equation.

$$\begin{aligned}\frac{dS}{dt} &= cR - \frac{aSI}{N} \\ \frac{dI}{dt} &= \frac{aSI}{N} - bI \\ \frac{dR}{dt} &= bI - cR\end{aligned}$$

We can use the solution of the system found by Monte Carlo simulations and fourth order Runge Kutta as a reference when training and developing our neural network. Initially our requirement is that the network is able to reliably reproduce the data we gathered from MC and RK4 earlier. As earlier, they should converge at the equilibrium state found through the analytical solution. After this we can reinitialize the neural network and let it find its own solution for weigth optimalization, using an objective function as the feedback parameter. [cost function == objective function?]

Reference: "<https://compphysics.github.io/MachineLearning/doc/pub/odenn/pdf/odenn-minted.pdf>"