

Machine Learning Engineer Nanodegree

Capstone Proposal

Kevin Siswandi
May 2020

Proposal

Domain Background

In the biosciences, [dynamic modeling](#) plays a very important role for understanding and predicting the behaviour of biochemical systems, with wide-ranging applications from bioengineering to precision medicine. For instance, understanding how a certain inhibitor affects the enzyme function is useful not just for improving titer, yield, and rate (TRY) in bioengineering, but also for designing personalized drugs. Traditionally, dynamic modeling of biochemical systems is done by painstakingly constructing a set of equations that characterises the system dynamics, which takes a long time to develop and does not scale with increasing amounts of data. To overcome these challenges, I will implement a data-driven method based on machine learning that allows faster development of predictive dynamics and improves in performance when more data are available. I am fascinated by the opportunities created by the abundance of data and computing power in bioinformatics and would love to contribute to shaping the future of bioscience with machine learning, one step at a time.

Reference: [MultiStep Neural Network](#)

Problem Statement

Following [MultiStep Neural Network](#), the objective is to find the function f that best represents the dynamics in the data. This is done by formulating a supervised learning problem embedded in the numerical framework of [linear multistep methods](#). I will build the multistep neural network with TensorFlow 2.0. RMSE will be used as the evaluation metric.

Datasets and Inputs

To generate time-series data for training and testing the model, I will simulate time-series concentrations from the [glycolytic oscillations in yeast cells](#). This can be done by solving the system of equations using a numerical integrator from `scipy`. I plan to use three set of parameter values for the ordinary differential equations of the dynamical system that will result in different qualitative behaviour of the system dynamics. I expect the multistep neural network to be robust enough to handle these three regimes.

The inputs are the simulated time-series data for yeast glycolysis. There is no further preprocessing because we already generate the data in regular time intervals.

Solution Statement

The MultiStep Neural Network will take in the time-series data as input and learns the function/derivative that describes the dynamics. Before it can be used to make predictions, the function must be integrated (using `scipy`). The benefit of this method is that it allows a full characterization of how the system will develop in time given only some initial values, which has plenty of use cases in bioengineering.

Benchmark Model

The classical benchmark model is the [implicit SINDy method](#), which uses LASSO regression to find a sparse representation to a library of candidate functions. Note: it's highly mathematical and assumes prior knowledge about the dynamical systems.

The goal of this project is to demonstrate that Multistep Neural Network can provide better predictive power than implicit SINDy, given more data.

Evaluation Metrics

Because we are dealing with continuous values, the natural metric to use is the root mean squared error (RMSE), which is the L2 norm of the discrepancy between predicted and true dynamics.

Project Design

The general workflow is: 1. Generation of training and test data from mathematical models (tools needed: `numpy`, `scipy`) 2. Construction of MultiStep Neural Network (tools needed: TensorFlow 2.0, `sympy`) 3. I will do step 1-2 for three regimes: sustained oscillation, damped oscillation, and state transition in yeast glycolysis. 4. Numerical integration of the dynamics (tools needed: `scipy`). 5. Determination of influential hyperparameters (step size, number of steps, multistep scheme, etc.) 6. Qualitative comparison in time-course plot and phase space.

I should remark that some terminologies used here are specific to the domain of systems biology and dynamical systems.