UPPSALA UNIVERSITET

Students:
Adam Lindell
Simon Strömstedt Hallberg

Supervisors:
Prashant Singh
Andreas Hellander

Black box time series modeling using recurrent neural networks

Aim

Generate a time series of a population count in systems biology using a neural network and only some underlying parameters of the system.

Background

The time series considered in systems biology depended on seven parameters and was simulated by a Gillespie algorithm but takes quite a bit of time . The task was to train a neural network to be able to produce an accurate simulated time series quicker than Gillespie. If this was the case then it could be used in greater extent and not slow down future code too much.

The problem

A time series is by definition events that follow each other in a particular order. Regular neural networks does not handle them that well, therefore recurrent neural networks were used in this project. They take previous values of a time series into account when predicting future values.

The project

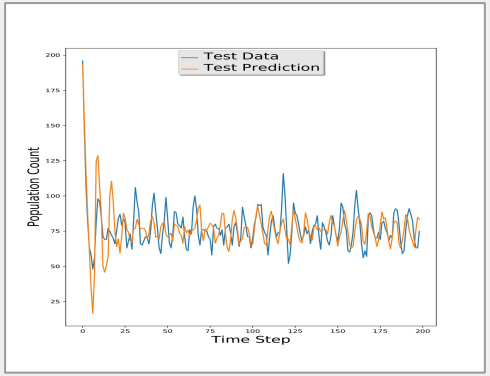
Make use of two different neural networks and high level programming libraries to train and predict time series. The project then became two distinct parts: LSTM in Python (Simon) and NARX in MATLAB (Adam).

Recurrent neural networks

Recurrent neural nets are used for predicting values in a series based on previous values.

Long Short Term Memory networks

Long short term memory nets are a type of recurrent neural nets that use different gates to control how the input affects the output.

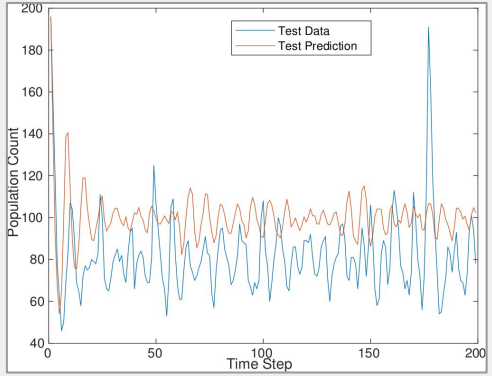


Results

Training the network on 125 time series each 200 time steps long. Root Mean Squared Error on best prediction: 12.49. Average Root Mean Squared Error on all time series: 43.11

Nonlinear Autoregressive Exogenous networks (NARX)

NARX produces an output y_i, and takes as input a value y_{i-1} and an input x, which is the exogenous part. The exogenous input was the parameters.



Results

Training the network on 125 time series each 200 time steps long. Absolute Mean Error on best prediction: 42.3433. Predicting 40 time series took approximately 0.2 seconds which means predicting only one would would be only a fraction of the time of the Gillespie.

Future work: Implement gaussian kernels. Experiment with different loss functions and normalization options. Use more data and sample it in different ways.