Lab work 9
Using Neural ODE for electricity consumption time series

Pankratov Viktor

Contents

1 Lab 9 report

1.1 Motivation

Many groups of natural processes are described by differential equations. Usually RNN are used to solve them. Another considered option is to solve the differential equation using neural networks, called Neural ODEs. Neural ODEs can also be used to generate trajectory according to initial observations. In this work the generation abilities of Neural ODEs is being tested using two network implementations.

1.2 Problem statement

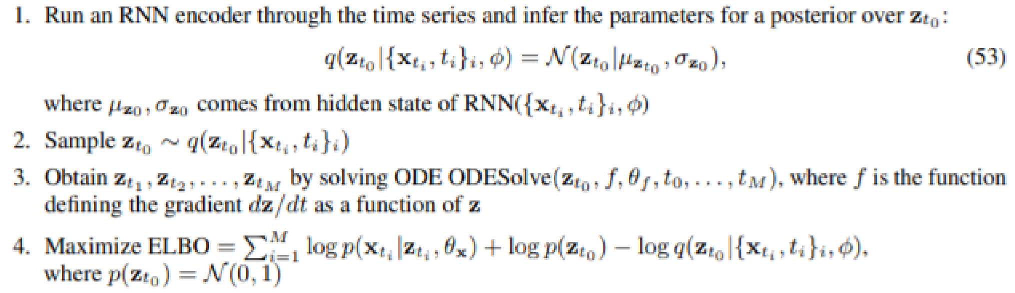
Given a sequence $z(t), t \leq T_0$ of time series elements. Each element consist of element value z and time t. A function f such as $\frac{\partial z}{\partial t} = f(z(t), t)$ is supposed to exist. The goal is to approximate f by neural network $\hat{f}(z(t), t, \theta)$ and find $z(t), T_0 \leq t \leq T_1$ for some T_1 by minimising

$$L(z(T_1)) = L(z(T_0) + \int_{T_0}^{T_1} \hat{f}(z(t), t, \theta) dt)$$

.

1.3 Solution

A model is crated to represent time series by a latent trajectory. Each trajectory is determined by initial state z_{T_0} and by latent dynamics shared across al the time series. Other states z_{t_i} are produced by ODE Solver.



1. Run an RNN encoder through the time series and infer the parameters for a posterior over $\mathbf{z}_{t_0}$:

$$q(\mathbf{z}_{t_0} | \{\mathbf{x}_{t_i}, t_i\}_i, \phi) = \mathcal{N}(\mathbf{z}_{t_0} | \mu_{\mathbf{z}_{t_0}}, \sigma_{\mathbf{z}_0}), \tag{53}$$

   where $\mu_{\mathbf{z}_0}, \sigma_{\mathbf{z}_0}$ comes from hidden state of RNN$(\{\mathbf{x}_{t_i}, t_i\}_i, \phi)$

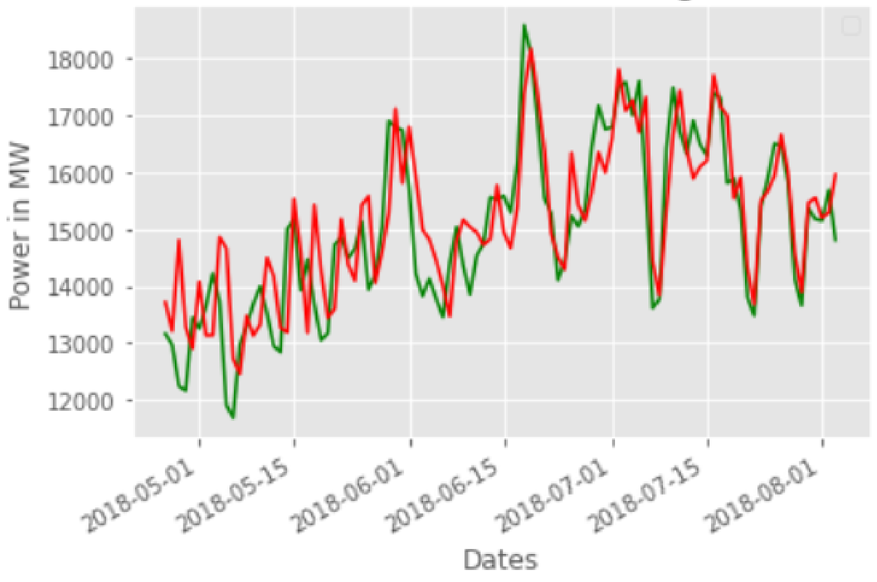
2. Sample $\mathbf{z}_{t_0} \sim q(\mathbf{z}_{t_0} | \{\mathbf{x}_{t_i}, t_i\}_i)$

3. Obtain $\mathbf{z}_{t_1}, \mathbf{z}_{t_2}, \ldots, \mathbf{z}_{t_M}$ by solving ODE ODESolve$(\mathbf{z}_{t_0}, f, \theta_f, t_0, \ldots, t_M)$, where f is the function defining the gradient $d\mathbf{z}/dt$ as a function of $\mathbf{z}$

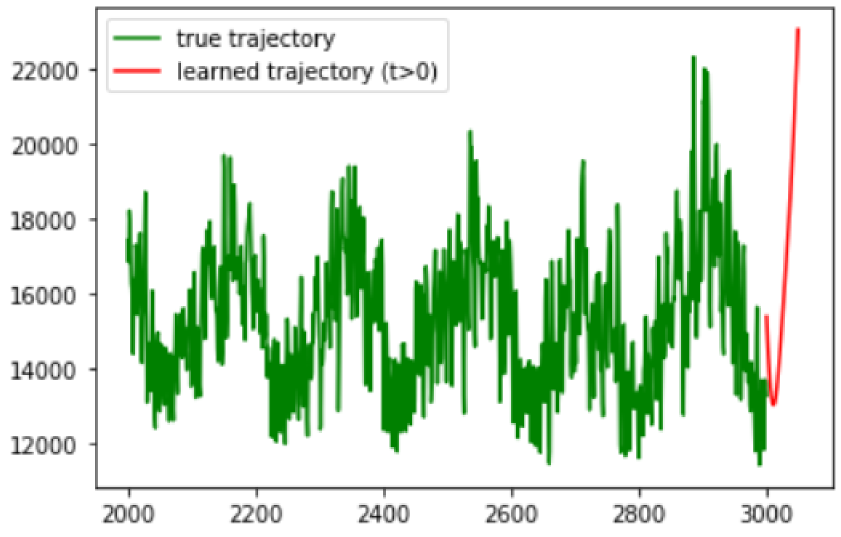
4. Maximize ELBO $= \sum_{i=1}^{M} \log p(\mathbf{x}_{t_i} | \mathbf{z}_{t_i}, \theta_{\mathbf{x}}) + \log p(\mathbf{z}_{t_0}) - \log q(\mathbf{z}_{t_0} | \{\mathbf{x}_{t_i}, t_i\}_i, \phi)$,
   where $p(\mathbf{z}_{t_0}) = \mathcal{N}(0, 1)$

1.4 Experimental results

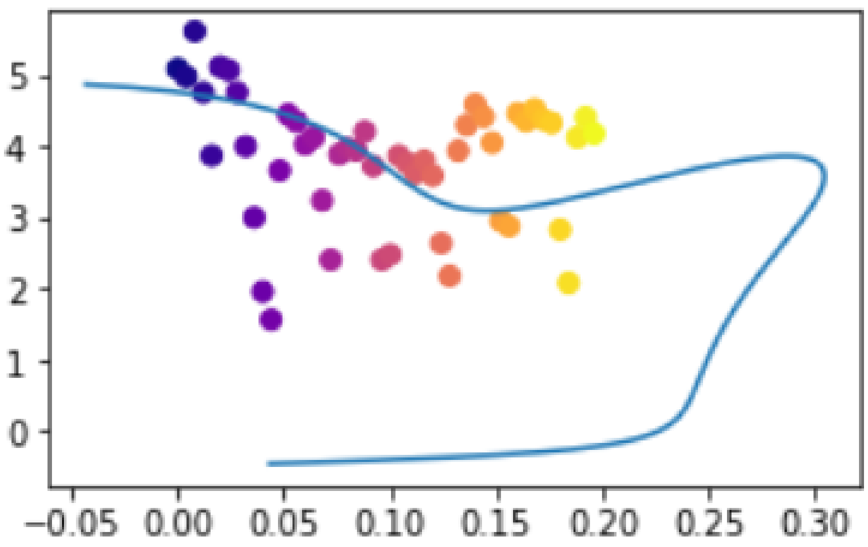
For experiments daily energy consumption data was used.



Two ODE Solvers implementations were used. First is used by authors of the original paper, the second is more simply written and easier to understand. However they are not exactly the same and they are compared:
First ODE results:



Second ODE results:



The first way is better at predicting locally while the second is better at trajectory direction predicting. However both of them give results worse than LSTM visually and in terms of metrics.

1.5 Useful links

The first ODE solver was taken from link . This and the other code parts are available on Github

1 Neural Ordinary Differential Equations. Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, David Duvenaud