Mathematical Methods of Forecasting

Yakovlev Konstantin

Contents

1 Lab work 6 **2**
1.1 Motivation 2
1.2 Problem statement 2
1.3 Problem solution 2
1.4 Code analysis 3
1.5 Computational experiment 3

1 Lab work 6

1.1 Motivation

The work investigates the problem of time series forecasting. The goal is to get a continuous forecast given discrete data points.

1.2 Problem statement

Given a time series $\mathbf{X} = (t_i, a_i)_{i=1}^{\ell}$, $t_1 \leq \ldots \leq t_\ell$. The number a_i is the accelerometer reading. Given also unobservable velocities $v_i \in \mathbb{R}^1$ at each timestamp t_i. The current velocity v_i can be computed through the previous one as follows: $v_i = f(v_{i-1}, a_i, \boldsymbol{\theta}_{\text{rnn}})$. It is assumed that at each interval $[t_{i-1}, t_i]$ the pendulum equation is fulfilled:

$$\begin{cases} \frac{d}{dt}a(t) = v(t), \\ \frac{d}{dt}v(t) = -\theta \sin a(t), \quad \theta > 0, \\ a(t_{i-1}) = a_{i-1} + \xi_{i-1}, \ v(t_{i-1}) = v_{i-1}. \end{cases} \qquad 2 \leq i \leq \ell$$

Let $\boldsymbol{\theta} = [\theta, \boldsymbol{\theta}_{\text{rnn}}]$. Given a loss function $\mathcal{L}(\boldsymbol{\theta}) = \sum_{i=2}^{\ell} \xi_i^2(\boldsymbol{\theta})$. Optimal parameters $\boldsymbol{\theta}$ are solution of the following optimization problem:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}).$$

1.3 Problem solution

Consider basic algorithm ODE-RNN [2]:

Algorithm 1 ODE-RNN

Require: Data points $\{(x_i, t_i)\}_{i=1}^{N}$.
  Initialize $\mathbf{h}_0$.
  for $i = 1, \ldots, N$ **do**
    $\mathbf{h}_i' = \text{ODESolve}(f_\theta, \mathbf{h}_{i-1}, (t_{i-1}, t_i))$.
    $\mathbf{h}_i = \text{RNNCell}(\mathbf{h}_i', x_i)$.
  end for
  For each i compute outputs $o_i = \text{OutputNN}(\mathbf{h}_i)$.
  return $\{o_i\}_{i=1}^{N}$

We introduce a modification of ODE-RNN. The difference is that we map hidden state to the space (from $\mathbb{R}^{\text{hidden}}$) of the ODE solution ($\mathbb{R}^2$) and back.

Algorithm 2 ODE-RNN with modifications

Require: Data points $\{(x_i, t_i)\}_{i=1}^{N}$.
  Initialize $\mathbf{h}_0$.
  for $i = 1, \ldots, N$ **do**
    $\mathbf{h}_i' = \text{ODESolve}(f_\theta, \text{To2d}(\mathbf{h}_{i-1}), (t_{i-1}, t_i))$.
    $\mathbf{h}_i = \text{RNNCell}(\text{ToHidden}(\mathbf{h}_i'), x_i)$.
  end for
  For each i compute outputs $o_i = \text{OutputNN}(\mathbf{h}_i)$.
  return $\{o_i\}_{i=1}^{N}$

We compute gradients of the loss function $\mathcal{L}(\boldsymbol{\theta})$ w.r.t. $\boldsymbol{\theta}$ using backpropagation. Then we perform optimizing step. After some iterations we get a solution.

1.4 Code analysis

The computational experiment can be found on GitHub repository[1].

1.5 Computational experiment

The goal of the computational experiment is to compare the prediction quality of the proposed method with other methods. The data was taken from WISDM dataset [1]. We considered an accelerometer measurements. The number of train timestamps is 60. The number of validation timestamps is 20. Hidden size of LSTM is 20. Output network, networks that map to hidden space and back are one layer dense networks.

First, we ran ODE-RNN with modifications with f_θ as a right hand side of a pendulum equation. We used Adam with learning rate $5 \cdot 10^{-3}$ for optimization. We ran 500 epochs of optimization. Validation loss is 0.6090.

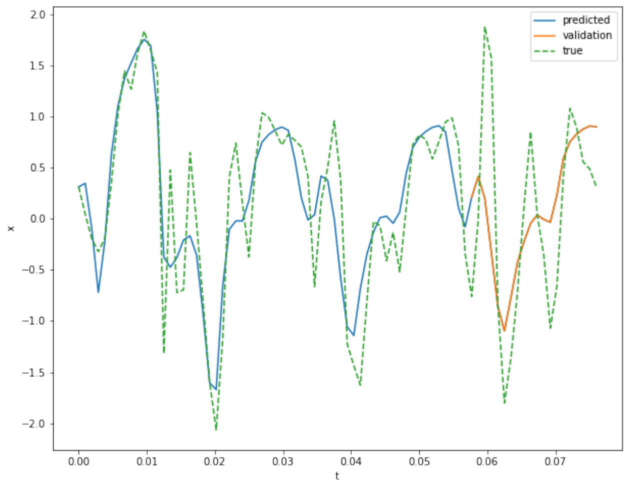

Figure 1: Forecast of ODE-RNN with modifications, pendulum equation.

After that we ran ODE-RNN with modifications with f_θ as a two fully connected layers with Tanh activation with hidden size 2. We also ran 500 epochs with Adam optimizer. The validation loss is 1.3758.

[1]https://github.com/Konstantin-Iakovlev/MathMethodsOfForecasting

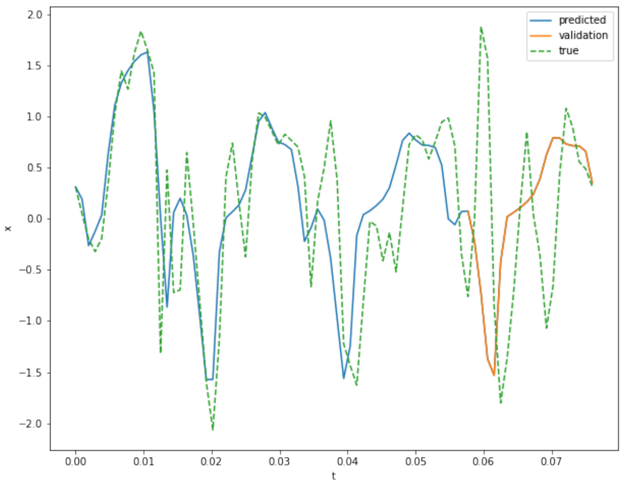

Figure 2: Forecast of ODE-RNN with modifications, dense network.

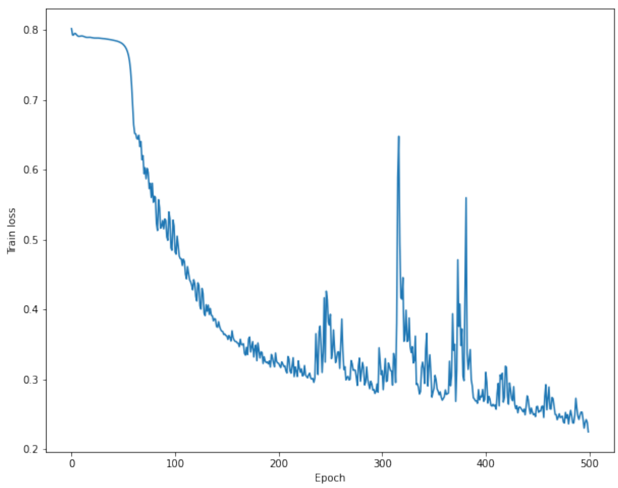

Figure 3: Train loss, pendulum equation.

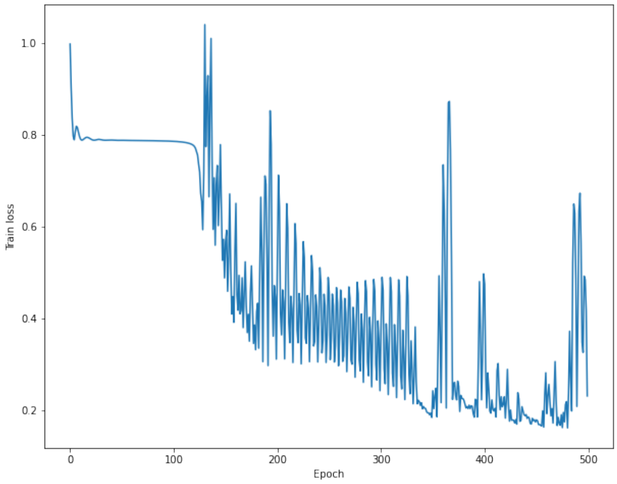

Figure 4: Train loss, dense network.

During our experiments, we noticed that the learning process of the network in the case of a pendulum is more stable and reach better minimum. It seems to be that the using of the pendulum equation's right hand side as a dynamic of an ODE is a powerful regularizer.

References

[1] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. "Activity recognition using cell phone accelerometers". In: *ACM SigKDD Explorations Newsletter* 12.2 (2011), pp. 74–82.

[2] Yulia Rubanova, Ricky T. Q. Chen, and David K Duvenaud. "Latent Ordinary Differential Equations for Irregularly-Sampled Time Series". In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: `https://proceedings.neurips.cc/paper/2019/file/42a6845a557bef704ad8ac9cb4461d43-Paper.pdf`.