Statistical Deep Learning (MT7042) - Project 3

Instructions

This project consists of only one task that should be solved individually. You are allowed to use any program package/library to complete your task. The report should be submitted on the course webpage as a single PDF file. The source code should be included as a separate script file.

Project Description

This project allows you to apply RNN with LSTM for time series forecasting.

Task 1

This project considers the time series of global surface temperature provided by NASA The data file `temperature_Anomaly_Data.txt` contains 142 time points. In the file, the first and second columns represent the year (from 1880 to 2021) and the global surface temperature anomaly (in Celsius), respectively. The temperature anomaly, referred to as *temperature* hereafter, is defined as the difference from the 1951-1980 average temperatures.

Your task is to train an RNN with LSTM to model the temperature time series for forecasting. Specifically, let the temperature time series be:

$$x(1), x(2), \ldots, x(n),$$

where n is the number of time points, $x(1)$ is the temperature for the year 1880, $x(2)$ is the temperature for the year 1881, and so on. A forecasting model takes in the temperature sequence:

$$\{x(t-2), x(t-1)\}$$

and predicts the temperature at the next time point $x(t)$.

Suggestive guidelines

These are suggested steps to help you complete the task. However, you are free to implement the modeling differently.

1. Time series forecasting is a supervised learning problem with predictor variables $\{..., x(t-2), x(t-1)\}$ and response variable $x(t)$. The question is how long the past sequence $\{..., x(t-2), x(t-1)\}$ one needs to predict the future.

2. Due to parameter sharing, RNN is not quite suitable to model non-stationary time series. Therefore, you may want to first transform the non-stationary temperature time series to a stationary one. One common (but not perfect) way to do so is to construct the difference time series $y(t)$ defined as:

$$y(t) = x(t) - x(t-1).$$

Note that the difference time series detrends the time series and usually results in short-term time correlation. In this case, you can simply use a short past sequence, say $(t-1)$ as predictor variable to predict the response variable $y(t)$.

3. Alternatively, work directly with the non-stationary series. In this case, you may need longer past sequences $\{..., x(t-2), x(t-1)\}$ as predictor variables. You can increase the length of past sequence from 1, 2, ... to see if the prediction gets better.

4. To perform validation, the time series should be divided into the training and test sets. Since we are dealing with time series, it is important that the training/test set contains a continuous segment of the time series. You can pick the training set as the data segment from year 1880 to 1980, and the test set as the data segment from year 1981 to 2021.

5. Temperature forecasting is a regression problem, so you may want to use the mean square error as your loss function.

6. Refer to examples of training LSTMs for time series forecasting:

   - example in R
   - example in Python

   NOTE: You may use part of the codes provided by the above pages, but please do not simply copy and paste the whole code for this project. Your code should be tailored to the time series forecasting in this project.

7. Write concisely, using bullet points, figures, and schematics. The report should not exceed three pages, excluding the code.

8. (Optional) Experiment with different model setups (e.g., RNN architecture, length of past sequence, parameter initialization). Vary one aspect at a time and compare results.

Problem 1

1. **Task 1:** Plot the time series $x(t)$ or $y(t)$ that you are going to model. Describe the predictor and response variables. (20 points)

2. **Task 2:** Describe and explain the RNN architecture you are training, e.g., number of time steps (τ), number of input units, hidden units, layers, activation function, and loss function. Is it a one-to-one or many-to-one or many-to-many architecture, etc.? Provide a model summary table. (30 points)

3. **Task 3:** Train the RNN with LSTM. We would not advise relying on automatic stopping criteria to determine the optimal model from the training curve. Include your code in a script. (30 points)

4. **Task 4:** Plot the training data set, the test data set and the predicted data for $x(t)$ as in the last figure in the R tutorial. If your prediction is good (e.g. can your prediction capture the rising trend and the oscillations in the time series from year 1981 to 2021?), discuss why this is so. If the prediction is not good, provide some discussion with reasoning on further improvements.(20 points)